



MOTOROLA
Semiconductor Products Inc.

ADVANCED SEMICONDUCTOR DEVICES (PTY) LTD
P.O. Box 2944, Johannesburg 2000
3rd Floor, Vegas House
123 Pritchard Street/Corner Mool Street
Johannesburg
Tel. No. 58-2856

AN-830
Application Note

AN INTELLIGENT TERMINAL WITH DATA LINK CAPABILITY

By
Charles Melear
Microprocessor Applications Engineer
Austin, Texas

INTRODUCTION

A small but powerful terminal complete with high speed data link can be constructed with a minimum number of NMOS LSI circuits. Operating systems can be developed to make this terminal act as a word processor, point-of-sale terminal, data input source, etc. The data link capability allows the operator to call in the resources of remote computers at synchronous serial data rates of up to 1.5 megahertz.

Five devices form the core of the terminal as shown in Figure 1. An MC6809 Microprocessor (MPU) was chosen because of its many hardware and software features.

The MC6845 CRT Controller (CRTC) permits the use of a video display monitor. This controller was chosen because it allows complete software control of the video display monitor. Vertical sync delay, horizontal sync width and delay, blanking, number of characters-per-row, and rows-per-screen are all programmable.

Serial keyboard input capability is provided by an MC6850 Asynchronous Communications Interface Adapter (ACIA) which performs the serial/parallel conversions. This application polls the ACIA to check for a data present indication instead of using an interrupt. This polling method provides the highest priority and shortest response time to the high speed data link.

High speed data link capabilities are provided by an MC6854 Advanced Data Link Controller (ADLC). The ADLC detects the start of a message, receives the message, calculates and appends a CRC character, and provides a closing flag. Serial data rates of 1.5 megahertz are possible with this system. To operate at these speeds, direct memory access capability is needed and is provided, in this application, by an MC6844 Direct Memory Access Controller (DMAC). A data transfer can be processed every four bus cycles when an MC6854 ADLC and an MC6844 DMAC are used together.

MC6845 CRT CONTROLLER (CRTC)

The CRTC provides horizontal sync, vertical sync, and blanking to a video display monitor along with the memory

address of the data to be displayed. A cursor output is also provided. Once the CRTC is initialized, it performs the function of controlling the video display monitor without intervention by the processor. Initialization is accomplished by writing the appropriate values into the 16 programmable registers. Figure 2 Sheet 1 is a worksheet which can be used to collect the information required to calculate the values needed for the CRTC register worksheet given in Figure 2 Sheet 2. It is assumed that the video display monitor uses a 60 hertz power source and a 15,750 hertz horizontal oscillator frequency. After initialization, the CRTC starts with the address located in the start address register. The ASCII character represented by the hexadecimal value at that location will appear in the upper left-hand corner of the video display monitor. The CRTC advances the memory address lines by one with each character clock. The first row will contain the number of characters specified in the horizontal display register.

Due to synchronization problems between the CRT clock and several other signals, it is possible that the first character could be only partially displayed. Figure 3 shows how this can happen because the time between the CRT clock and display enable (Tx) is an internal function of the CRTC. The first character will be partially displayed because display enable goes high approximately in the middle of the first character. This problem can be resolved by writing an ASCII blank (20) at the first character location and using the second character location to display the first character.

The screen memory must be accessible to the processor for updating. Since the CRTC memory address lines normally drive the screen memory, multiplexers are used to select either the CRTC memory address lines or the processor address lines. A decoding network selects the processor address lines any time an address between \$0000 and \$1FFF is detected. The data bus for the screen memory is isolated from the processor data bus by SN74LS243 transceivers. These devices are normally in the high-impedance state in both directions except during a processor read or write of the

Format Worksheet			
1.	Displayed Characters Per Row	_____	Char.
2.	Displayed Character Rows Per Screen	_____	Rows
3.	Character Matrix	a. Columns _____	Columns
		b. Rows _____	Rows
4.	Character Block	a. Columns _____	Columns
		b. Rows _____	Rows
5.	Frame Refresh Rate	_____	Hz
6.	Horizontal Oscillator Frequency	_____	Hz
7.	Active Scan Lines (Line 2 × Line 4b)	_____	Lines
8.	Total Scan Lines (Line 6 + Line 5)	_____	Lines
9.	Total Rows Per Screen (Line 8 + Line 4b)	_____	Rows
9a.	Number of Scan Lines Remaining From Line 9	_____	Lines
10.	Vertical Sync Delay (Character Rows)	_____	Rows
11.	Vertical Sync Width (Scan Lines)	16	Lines
12.	Horizontal Sync Delay (Character Times)	_____	Char. Time
13.	Horizontal Sync Width (Character Times)	_____	Char. Time
14.	Horizontal Scan Delay (Character Times)	_____	Char. Time
15.	Total Character Times (Line 1 + 12 + 13 + 14)	_____	Char. Time
16.	Character Rate (Line 6 times 15)	_____	Hz
17.	Dot Clock Rate (Line 4a times 16)	_____	Hz

Figure 2. CRTC Programming Worksheet (Sheet 1 of 2)

CRTC Register Worksheet			
		Decimal	Hex
R0	Horizontal Total (Line 15 minus 1)	_____	_____
R1	Horizontal Displayed (Line 1)	_____	_____
R2	Horizontal Sync Position (Line 1 + Line 12)	_____	_____
R3	Horizontal Sync Width (Line 13)	_____	_____
R4	Vertical Total (Line 9 minus 1)	_____	_____
R5	Vertical Adjust (Line 9a Lines)	_____	_____
R6	Vertical Displayed (Line 2)	_____	_____
R7	Vertical Sync Position (Line 2 + Line 10)	_____	_____
R8	Interlace (00 Normal, 01 Interlace, 03 Interlace and Video)	_____	_____
R9	Max Scan Line Add (Line 4b minus 1)	_____	_____
R10	Cursor Start	_____	_____
R11	Cursor End	_____	_____
R12	Start Address (H)	_____	_____
R13	Start Address (L)	_____	_____
R14	Cursor (H)	_____	_____
R15	Cursor (L)	_____	_____
R16	Light Pen (H)	_____	_____
R17	Light Pen (L)	_____	_____

Figure 2. CRTC Programming Worksheet (Sheet 2 of 2)

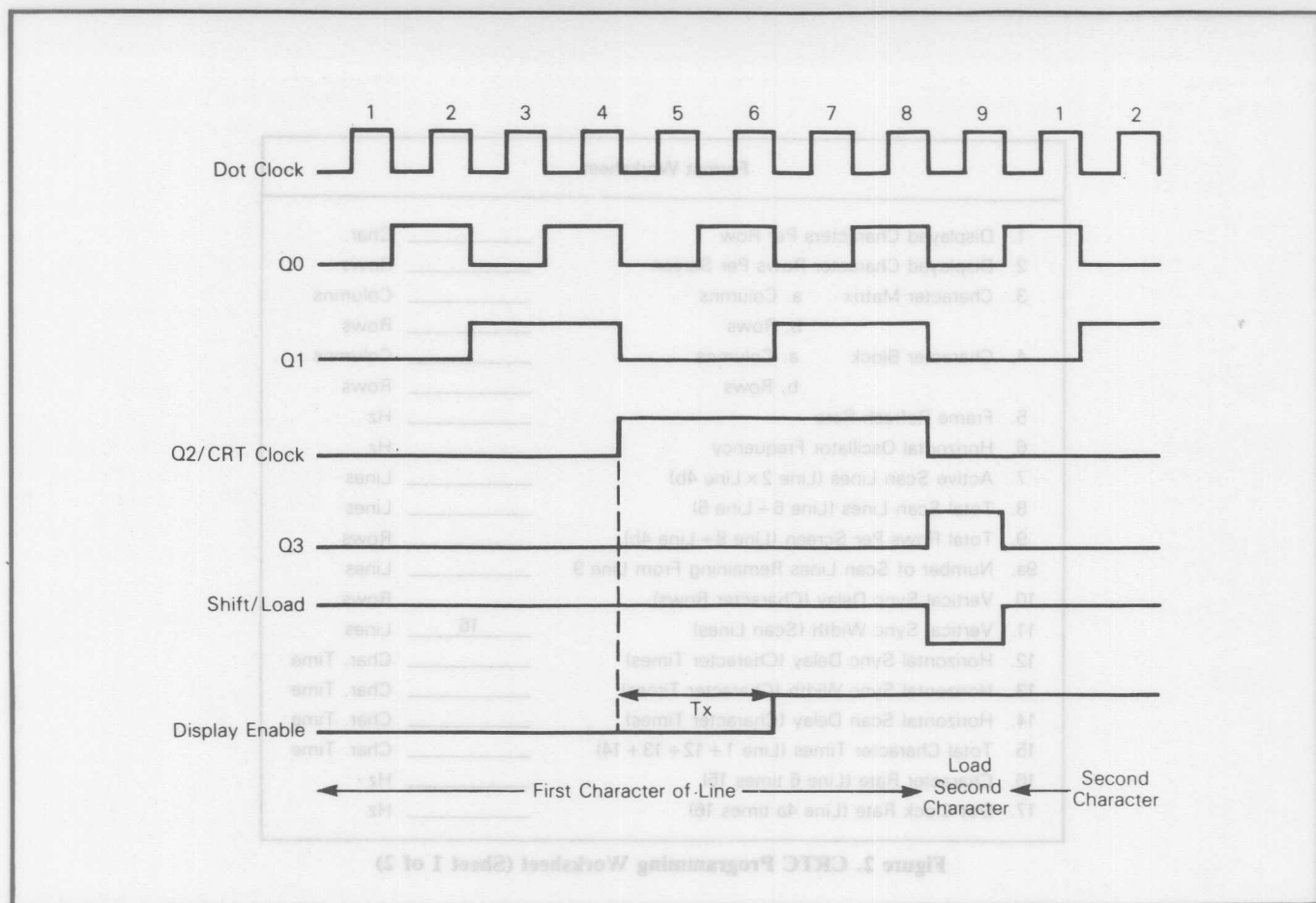


Figure 3. First Character Timing

the first character row, increments the row address by one, and then steps through the same addresses again. This procedure is repeated until the row address is equal to the address contained in the maximum scan line address register. The row address is reset to zero and the second character row is displayed.

A cursor may be programmed to appear at any location within the display memory area. The cursor output signal is logically ORed with the output of the data output shift register to form a new signal called cursor plus data.

Display enable is used for vertical and horizontal blanking. The data to the video display monitor must be enabled only during the time that the beam of the video display monitor is sweeping what has been defined as the display area. Otherwise, random data may appear at the edges of the screen and horizontal and vertical retrace lines may also be visible. Display enable goes high as the first character of a row is displayed and goes low just after the last character of a row is displayed.

Signals which include cursor plus data, display enable, and the select signal to the address line multiplexers are ANDed to form the composite data signal applied to the video display monitor. The select signal to the address line multiplexer is included to suppress any spurious data that may occur when the processor accesses the display memory. Composite data is fed to a D-type flip-flop that is clocked by the dot clock. This ensures that boundaries between dot periods in the composite data signal occur at regular intervals.

MC6844 DIRECT MEMORY ACCESS CONTROLLER (DMAC)

This application has local keyboard interface capability through use of an MC6850 Asynchronous Communications Interface Adapter (ACIA). It also has serial data link capability through the use of an MC6854 Advanced Data Link Controller (ADLC). This is a high speed data link capable of data transfer rates up to 1.5 megabits per second. If used at maximum speed in full duplex, a polling routine would not be able to handle the transmitted and received data. Therefore, direct memory access capability is needed. At one megabit data transfer rates, a data transfer must occur every four microseconds if full duplex operation is used. An MC6844 Direct Memory Access Controller (DMAC) can transfer data at that rate. One transfer is made every eight microseconds on each of two channels or one byte received and one byte transmitted during eight microseconds. The DMAC has four channels, but only two are used in this application. When enabled, two different pins on the ADLC are used to indicate that the transmit data register is empty and that the receiver FIFO buffer is full. These signals are used to make transfer requests to channels zero and one of the DMAC.

When the transfer request line (TxRQ) goes high in response to a service request from the ADLC, the DMAC requests the data bus from the MPU. When the data and address buses are available, the MPU will assert bus available (BA) and bus status (BS). The logical AND of these signals is the DMA grant signal (DGRNT) to the DMAC. When DMA

grant is received, the DMAC automatically takes control of the buses in one cycle and performs the data transfer during the next cycle. The bus request from the DMAC is released during the transfer cycle. The MC6809 will not attempt to regain the bus until one full cycle after the release of bus request. The bus available and bus status signals from the MPU are released immediately after the removal of bus requests which causes DMA grant to go low. This allows the DMAC to put its bus drivers in the high-impedance state in the cycle following the transfer without the possibility of bus contention by the MPU.

The DMAC has a number of 8-bit registers to be programmed. Figure 4 is an illustration of these registers. Channel zero is a transmit channel and its address register (registers 0 and 1) is loaded with the first address in memory to be transferred. The channel zero byte count register (registers 2 and 3) is loaded with the number of bytes to be transferred. The address register for channel one (register 4 and 5) is loaded with the first address in memory to serve as a destination for data. The byte count register for channel one (registers 6 and 7) should be loaded with \$FFFF since the length of an incoming message is generally not known. This value will allow

a message of any length. Registers 8 through F are not used. In this application, channel zero is programmed for the three-state control steal transfer mode and read (from memory to ADLC); and channel one is programmed for three-state control steal transfer mode and write (from ADLC to memory).

The priority control register is used to enable the transmit and receive channels when desired. Only interrupt request/DMA end ($\overline{\text{IRQ}}/\overline{\text{DEND}}$) for channel zero is enabled in the interrupt control register. This will cause an interrupt when the channel zero byte count register is decremented to zero indicating that all bytes have been transferred. A DMA end ($\overline{\text{DEND}}$) will occur when the last byte is transferred to the transmit register of the ADLC. DMA end ($\overline{\text{DEND}}$) and interrupt request ($\overline{\text{IRQ}}$) are multiplexed on one pin. By taking the logical OR of $\overline{\text{DGRNT}}$ and $\overline{\text{IRQ}}/\overline{\text{DEND}}$, a separate $\overline{\text{IRQ}}$ can be obtained. The separate $\overline{\text{DEND}}$ is obtained by taking the logical OR of the transfer strobe (TxSTB) and interrupt request/DMA end ($\overline{\text{IRQ}}/\overline{\text{DEND}}$). In actual use the DMAC is programmed and enabled before the ADLC is enabled. This will ensure that transfers can begin immediately upon initialization of the ADLC.

Programming Model

Register	Address (Hex)	Register Content							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Control	1x*	DMA End Flag (DEND)	Busy/Ready Flag	Not Used	Not Used	Address Up/Down	TSC/Halt	Burst/Steal	Read/Write (R/W)
Priority Control	14	Rotate Control	Not Used	Not Used	Not Used	Request Enable #3 (RE3)	Request Enable #2 (RE2)	Request Enable #1 (RE1)	Request Enable #0 (RE0)
Interrupt Control	15	DEND IRQ Flag	Not Used	Not Used	Not Used	DEND IRQ Enable #3 (DIE3)	DEND IRQ Enable #2 (DIE2)	DEND IRQ Enable #1 (DIE1)	DEND IRQ Enable #0 (DIE0)
Data Chain	16	Not Used	Not Used	Not Used	Not Used	Two/Four Channel Select (2/4)	Data Chain Channel Select B	Data Chain Channel Select A	Data Chain Enable

*The x represents the binary equivalent of the channel desired.

Address and Byte Count Registers

Register	Channel	Address (Hex)
Address High	0	0
Address Low	0	1
Byte Count High	0	2
Byte Count Low	0	3
Address High	1	4
Address Low	1	5
Byte Count High	1	6
Byte Count Low	1	7
Address High	2	8
Address Low	2	9
Byte Count High	2	A
Byte Count Low	2	B
Address High	3	C
Address Low	3	D
Byte Count High	3	E
Byte Count Low	3	F

Figure 4. MC6844 DMAC Registers

MC6854 ADVANCED DATA LINK CONTROLLER (ADLC)

The ADLC handles the data link protocol. Basically, the ADLC transmits and receives serial data in full duplex. The data format of message frame is shown in Figure 5. When transmitting, the transmit data output (TxD) will either be high (mark idle) or sending a series of opening flags (flag idle). Upon writing a word to the transmit FIFO register, an opening flag will be sent followed by the data. Details of the ADLC registers are given in Figure 6. Data must be supplied to the transmit FIFO register at a rate sufficient to ensure that the data output shift register never becomes empty. The last byte to be transmitted is written to the transmit last data register. As soon as the last byte is transmitted, the ADLC automatically appends a 16-bit cyclic redundancy character (CRC) in the frame check sequence field and a closing flag. The receiver constantly searches the data stream for an opening flag with which to synchronize. After an opening flag is detected, the first non-flag character and all succeeding bytes are shifted into the receiver FIFO register and CRC calculation is started. The receiver FIFO register must be read fast enough to ensure that a receiver overrun does not occur. When a closing flag is detected, the ADLC takes the prior 16 bits and compares it to the CRC generated by the receiver. The CRC is not shifted into the receiver FIFO register.

The chip select (\overline{CS}) pin of the ADLC must be asserted whenever the DMAC requests data by issuing a transmit strobe or when the address of the ADLC appears on the address bus. The logical ANDing of TxSTB and the address of the ADLC is used to develop a composite chip select (\overline{CS}) signal.

When a DMA transfer occurs between memory and the ADLC, the DMAC controls the R/\overline{W} line for the system. During the transfer cycle, the R/\overline{W} line of the ADLC must be inverted with respect to the system R/\overline{W} line. This is accomplished by exclusive ORing TxSTB and R/\overline{W} . If TxSTB is low (no transfer), the output follows R/\overline{W} . If TxSTB is high (transfer cycle), the output is the complement of R/\overline{W} .

The ADLC requires that the last byte to be transferred be treated differently. The system may set bit four of control register two high and write the last byte into the transmitter (continue) data register or the last byte can be written into the

transmitter (last) data register. In this application, the latter method is used by using TxSTB, $\overline{IRQ/DEND}$, and R/\overline{W} to control a dual, 4-to-1 data selector. The truth table for the data selector is shown in Table 1. When DEND is low, the DMAC is indicating that this is the last byte. DEND occurs coincidentally with TxSTB which forces the register selects (RS0, RS1) of the ADLC high and selects the transmitter (last) data register. If only TxSTB is low, register select zero will be low and register select one will be high and the transmitter (continue) data register will be selected.

MC6809 MICROPROCESSING UNIT (MPU)

The MC6809 must be discussed from two viewpoints — hardware and software.

HARDWARE — The internal clock of the MPU is made to work with the MC6844 DMA Controller. Figure 7 is a timing diagram for a DMA response and three-state steal. The DMA request three-state control steal (\overline{DRQT}) output of DMAC drives the $\overline{DMA/BREQ}$ input of the MPU. As shown in Figure 7, the first full cycle following \overline{DRQT} going low (which causes $\overline{DMA/BREQ}$ to also go low) is a dead cycle. Since the \overline{DRQT} low output from the DMA results in the $\overline{DMA/BREQ}$ input to MC6809 going low, it is a dead cycle for both the DMA and the MC6809. Dead time is the time required for the MPU to relinquish control of the bus and the DMAC to gain control of the bus. The next cycle accommodates the DMA transfer. During this cycle, \overline{DRQT} is released. The MPU automatically inserts one dead cycle after $\overline{DMA/BREQ}$ is released. This gives the DMAC one cycle to relinquish control of the bus and the MPU to gain control of the bus. After the dead cycle, the MPU assumes normal control.

The MC6809 has no equivalent of the valid memory address (VMA) signal which is available on the MC6800. Normally a VMA is not needed; however, during the dead cycles which precede and follow a DMA transfer, the buses are undefined. This allows the possibility of a spurious write into a random memory location. This possibility can be eliminated by developing a signal called direct memory access valid memory address (DMAVMA). The DMA grant DGRNT signal from the DMAC and the E signal are used to

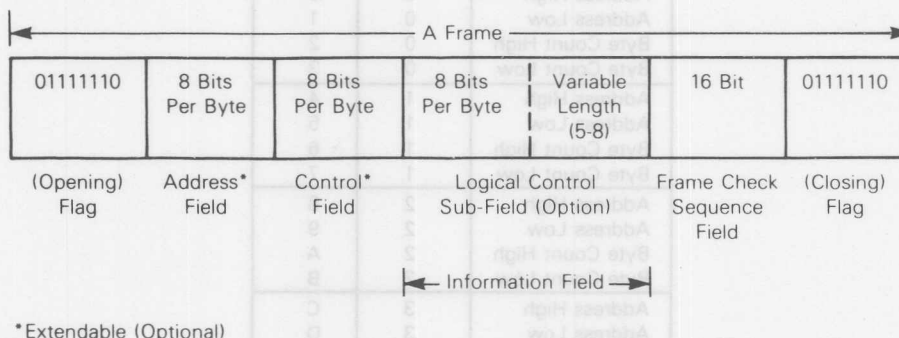


Figure 5. Data Format of a Message Frame

Read Only Registers	Bit #	RS1 RS0 = 00	RS1 RS0 = 01	RS1 RS0 = 10	RS1 RS0 = 11	
		Status Register #1	Status Register #2	Receiver Data Register		
	0	RDA	Address Present	Bit 0	Same as RS1, RS0 = 10	
	1	Status #2 Read Request	Frame Valid	Bit 1		
	2	Loop	Inactive Idle Received	Bit 2		
	3	Flag Detected (When Enabled)	Abort Received	Bit 3		
	4	CTS	FCS Error	Bit 4		
	5	Tx Underrun	DCD	Bit 5		
	6	TDRA/Frame Complete	Rx Overrun	Bit 6		
	7	IRQ Present	RDA (Receiver Data Available)	Bit 7		

Write Only Registers	Bit #				Transmitter Data	Transmitter Data	
		Control Register = 1	Control Register = 2 (C1b0 = 0)	Control Register = 3 (C1b0 = 1)	(Continue Data)	(Last Data) (C1b0 = 0)	Control Register = 4 (C1b0 = 1)
	0	Address Control (AC)	Prioritized Status Enable	Logical Control Field Select	Bit 0	Bit 0	Double Flag Single Flag Interframe Control
	1	Receiver Interrupt Enable (RIE)	2 Byte/1 Byte Transfer	Extended Control Field Select	Bit 1	Bit 1	Word Length Select Transmit #1
	2	Transmitter Interrupt Enable (TIE)	Flag/Mark Idle	Auto, Address Extension Mode	Bit 2	Bit 2	Word Length Select Transmit #2
	3	RDSR Mode (DMA)	Frame Complete/ TDRA Select	01/11 Idle	Bit 3	Bit 3	Word Length Select Receive #1
	4	TDSR Mode (DMA)	Transmit Last Data	Flag Detected Status Enable	Bit 4	Bit 4	Word Length Select Receive #2
	5	Rx Frame Discontinue	CLR Rx Status	Loop/Non-Loop Mode	Bit 5	Bit 5	Transmit Abort
	6	Rx RESET	CLR Tx Status	Go Active on Poll/Test	Bit 6	Bit 6	Abort Extend
	7	Tx RESET	RTS Control	Loop On-Line Control DTR	Bit 7	Bit 7	NRZI/NRZ

Figure 6. MC6854 ADLC Internal Register Details

Table 1. Data Selector Truth Table

Operation	TxSTB	DEND	R/W	A	B	1Y	2Y
Normal Operation No DMA Transfer	1	X	X	1	1	A1	A0
DMA Transfer from ADLC to Memory	0	1	0	0	1	1	0
DMA Transfer from Memory to ADLC	0	1	1	0	1	1	0
DMA Transfer of Last Byte from Memory to ADLC	0	0	0	0	0	1	1
DMA Transfer of Last Byte from ADLC to Memory	0	0	1	0	1	1	0

develop the $\overline{\text{DMAVMA}}$ signal as shown in Figure 8. A timing diagram showing the effect of the $\overline{\text{DMAVMA}}$ signal is given in Figure 7.

SOFTWARE — The flowchart used to generate the software to operate this system is shown in Figure 9 and the software listings are shown in Figures 10 and 11. Figure 10 uses the ADLC in the priority mode while Figure 11 uses the non-priority mode. The software overhead in this program limits the operation of the data link to about 62 kHz. However, this program is highly instructive in the use of the ADLC/DMA combination.

The MC6809 has been shown to have definite hardware advantages primarily due to the internal DMA compatible clocks; however, the software advantages are also quite

impressive. The use of the Direct Page Register allows significant reductions in the amount of object code that must be generated. The Direct Page Register is an 8-bit register that forms the upper byte of a 16-bit address instead of assuming the upper byte is \$00 when a direct instruction is executed.

The two programs contained in Figures 10 and 11 show the use of the ADLC in the priority and non-priority modes, respectively. The priority mode program requires 660 bytes of code without the Direct Page Register. However, 43 bytes of code were saved when the scratch RAM was moved from \$0000 to \$BF10. The Direct Page Register was set to \$BF so that the scratch RAM and peripherals could all be addressed with direct instructions. The non-priority mode, which originally required 718 bytes of code, was reduced by 36 bytes when the Direct Page Register was used.

By setting the Direct Page Register to \$BF, one byte of code will be saved each time locations \$BF00 to \$BFFF are accessed. An extended instruction takes three bytes of code as opposed to two for direct. For programs that must operate on real time events, this also has the advantage of executing a memory access in one less clock cycle. The program operating this system limits the serial transmission rate due to software overhead. By reducing this overhead, the hardware can operate faster.

Another advantage is position independency. Conditional branches of ± 32768 bytes can be executed. This covers the entire memory address space available to the MC6809. Since branches are program counter relative, this makes them independent of where the program originates. MC6800

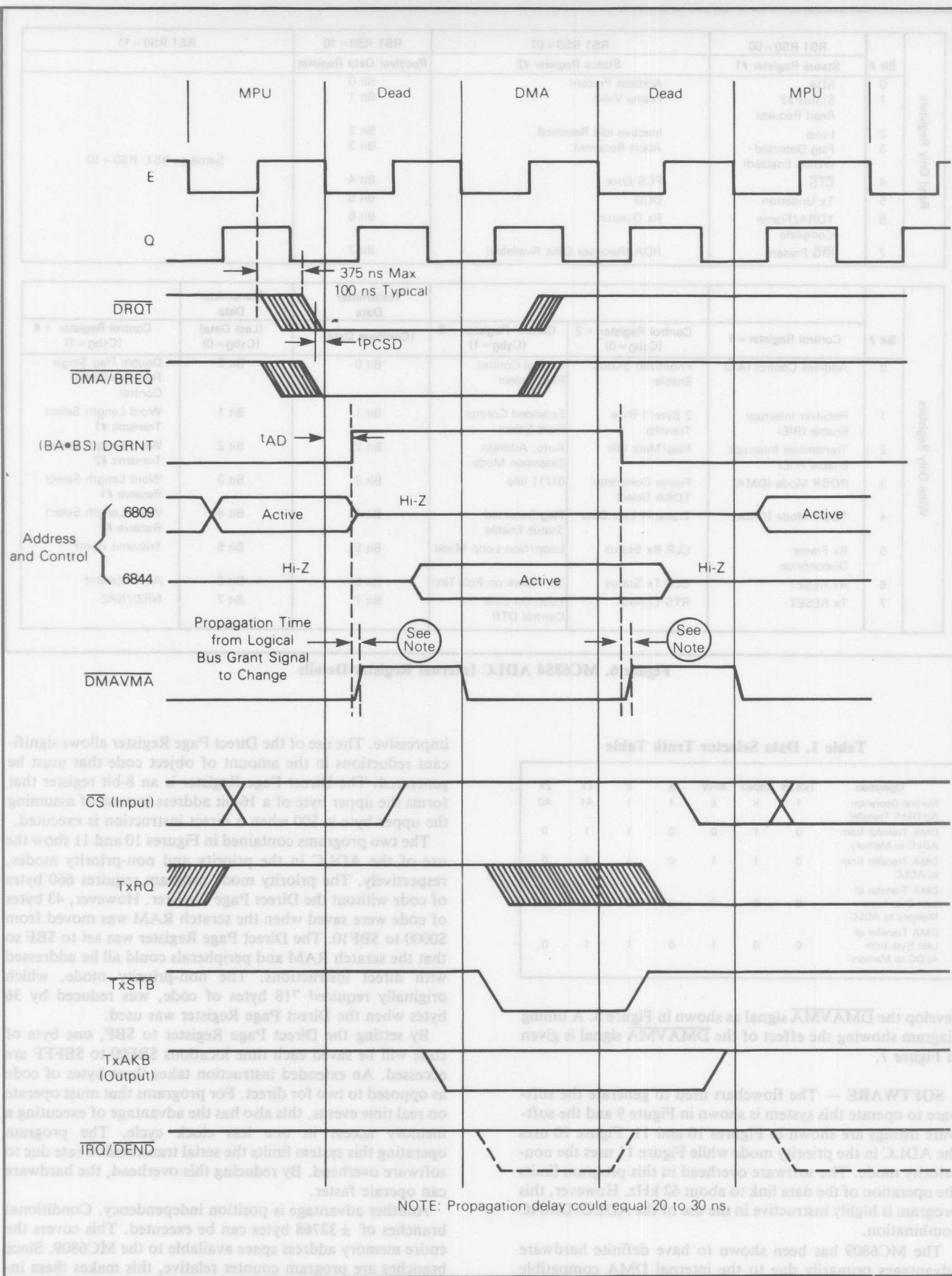


Figure 7. Three-State Steal DMA Timing

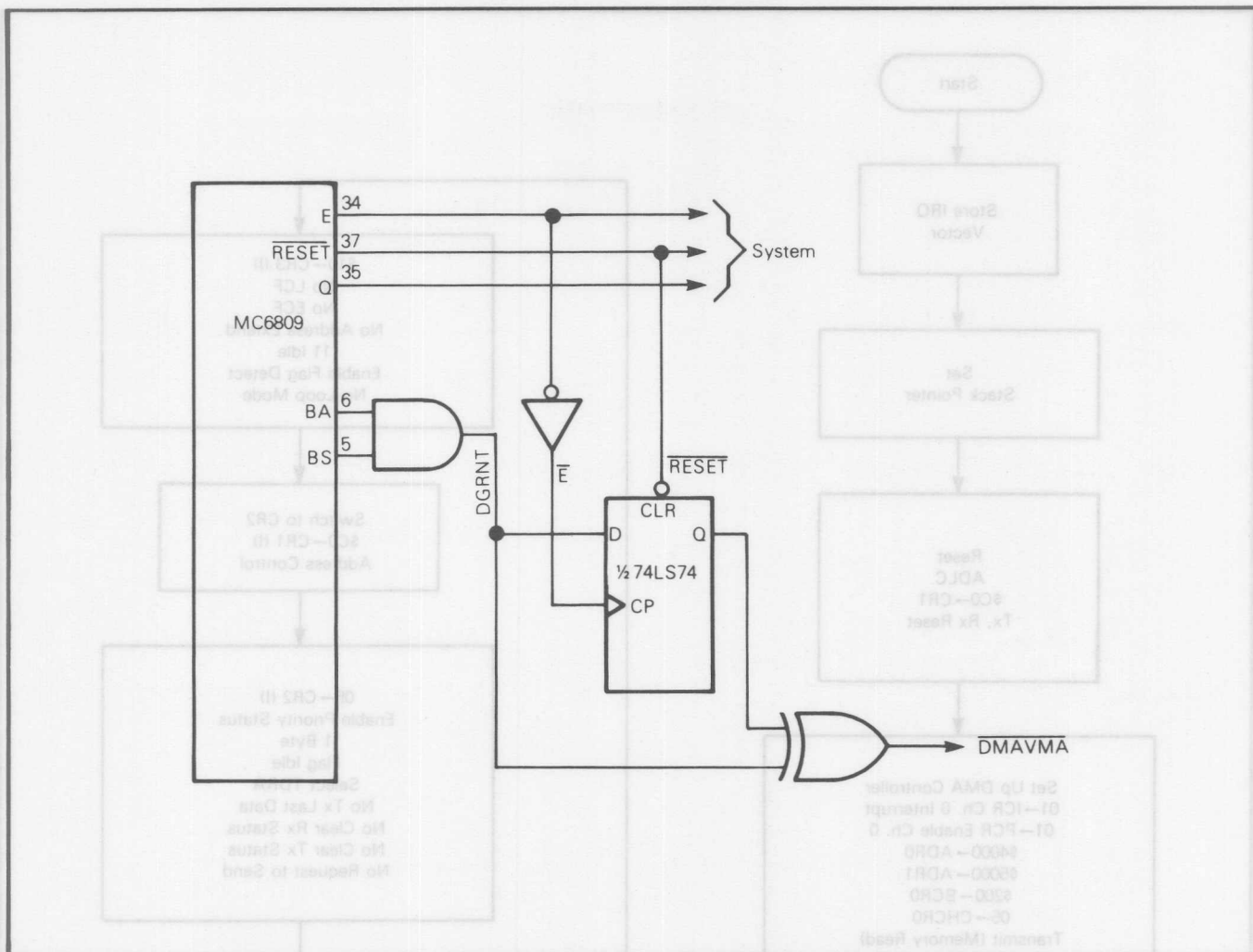


Figure 8. DMAVMA Generation Circuit

branches being ± 128 bytes must be used to branch to jump statements for moves of greater than 128 bytes. A long branch to subroutine instruction also enables $\pm 32K$ branches which are also program counter relative. When a program is completely position independent, the code can be placed anywhere in the memory space and work. MC6800 programs cannot be made position independent unless they

are written in the first 256 memory locations. Therefore, the MC6809 offers the convenience of position-independent ROMs.

SYSTEM SCHEMATIC

Figure 12 is the schematic for the intelligent terminal.

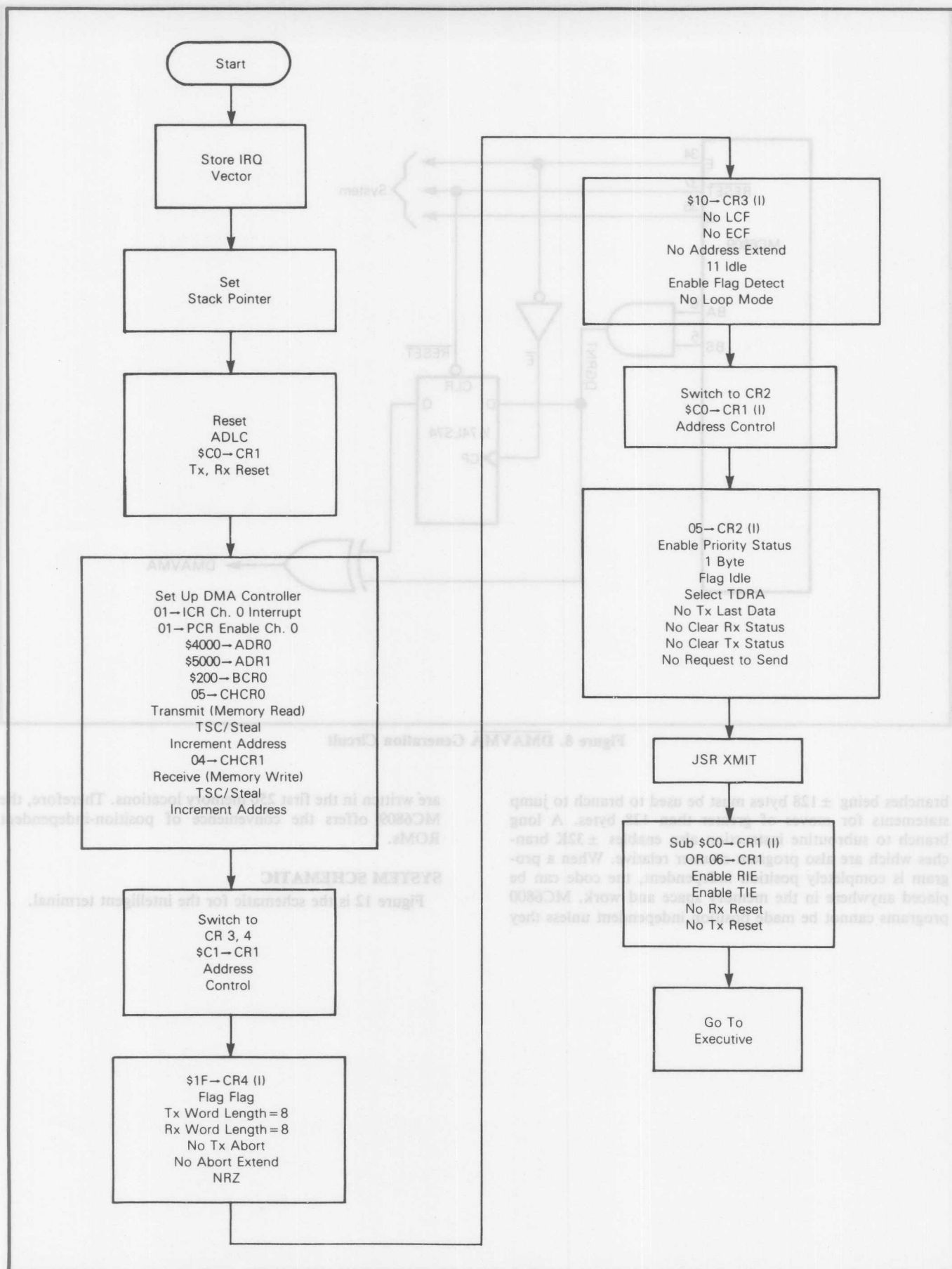


Figure 9. Flowchart of DMA-ADLC Program (Sheet 1 of 9)

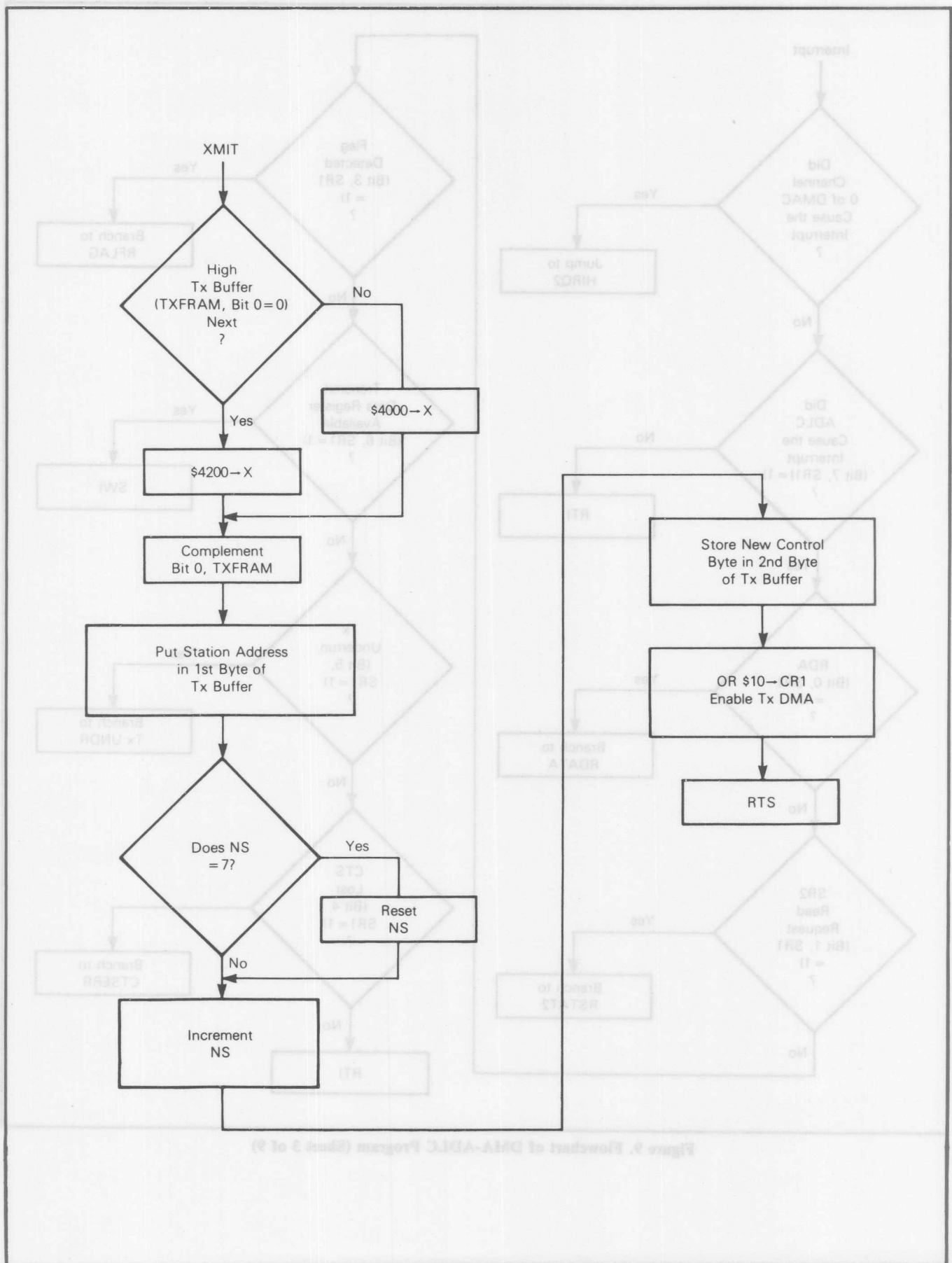


Figure 9. Flowchart of DMA-ADLC Program (Sheet 2 of 9)

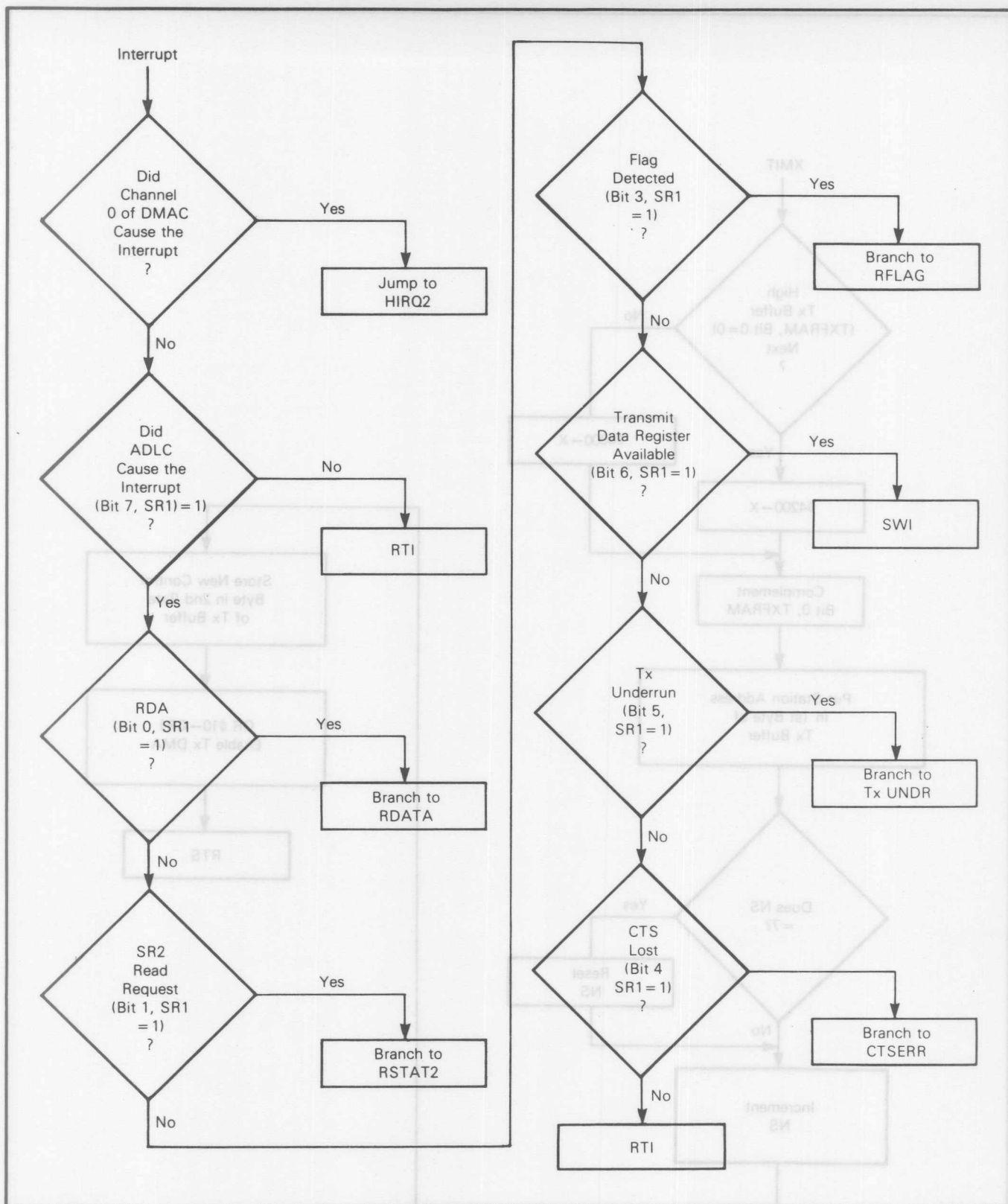


Figure 9. Flowchart of DMA-ADLC Program (Sheet 3 of 9)

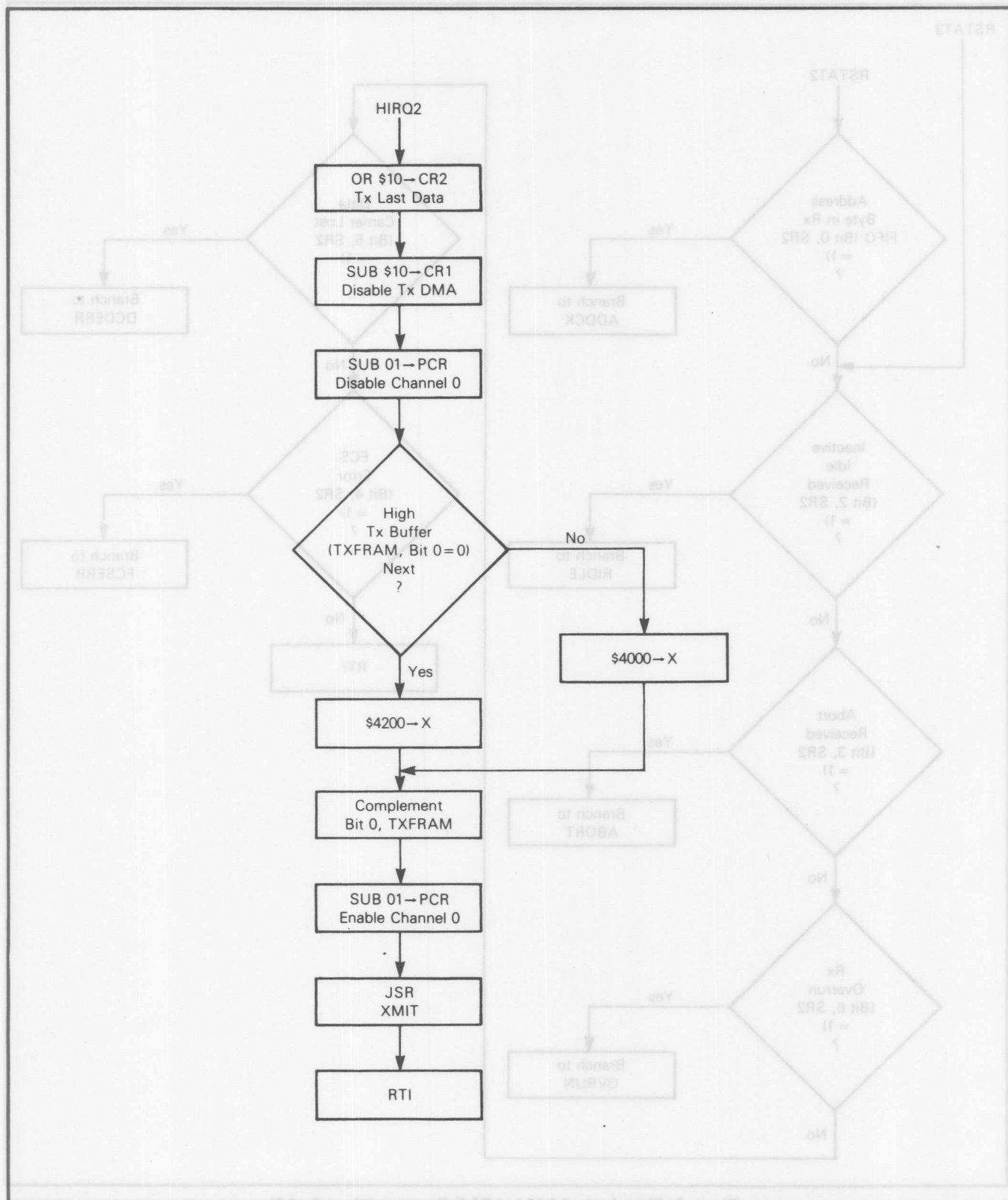


Figure 9. Flowchart of DMA-ADLC Program (Sheet 4 of 9)

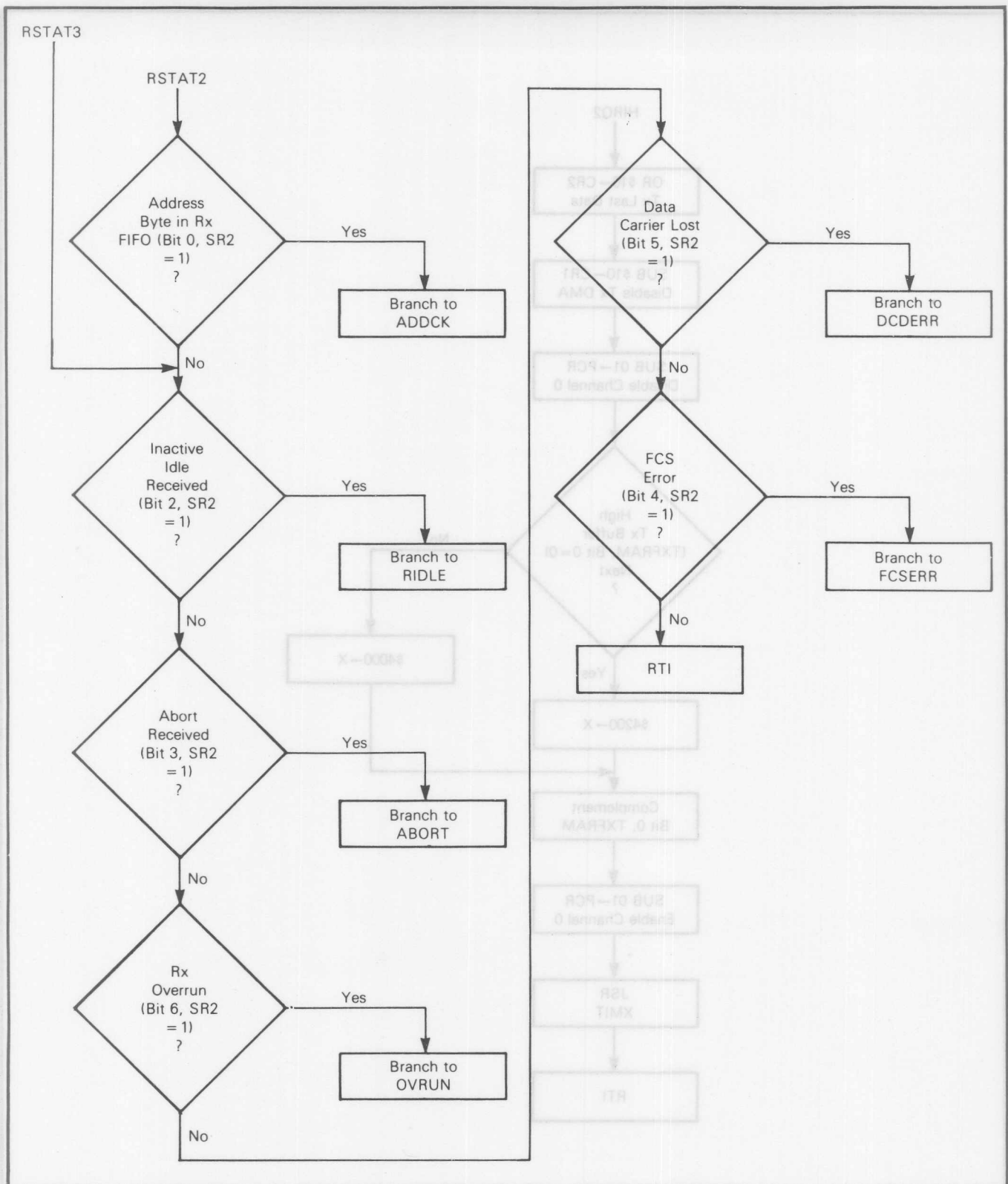


Figure 9. Flowchart of DMA-ADLC Program (Sheet 5 of 9)

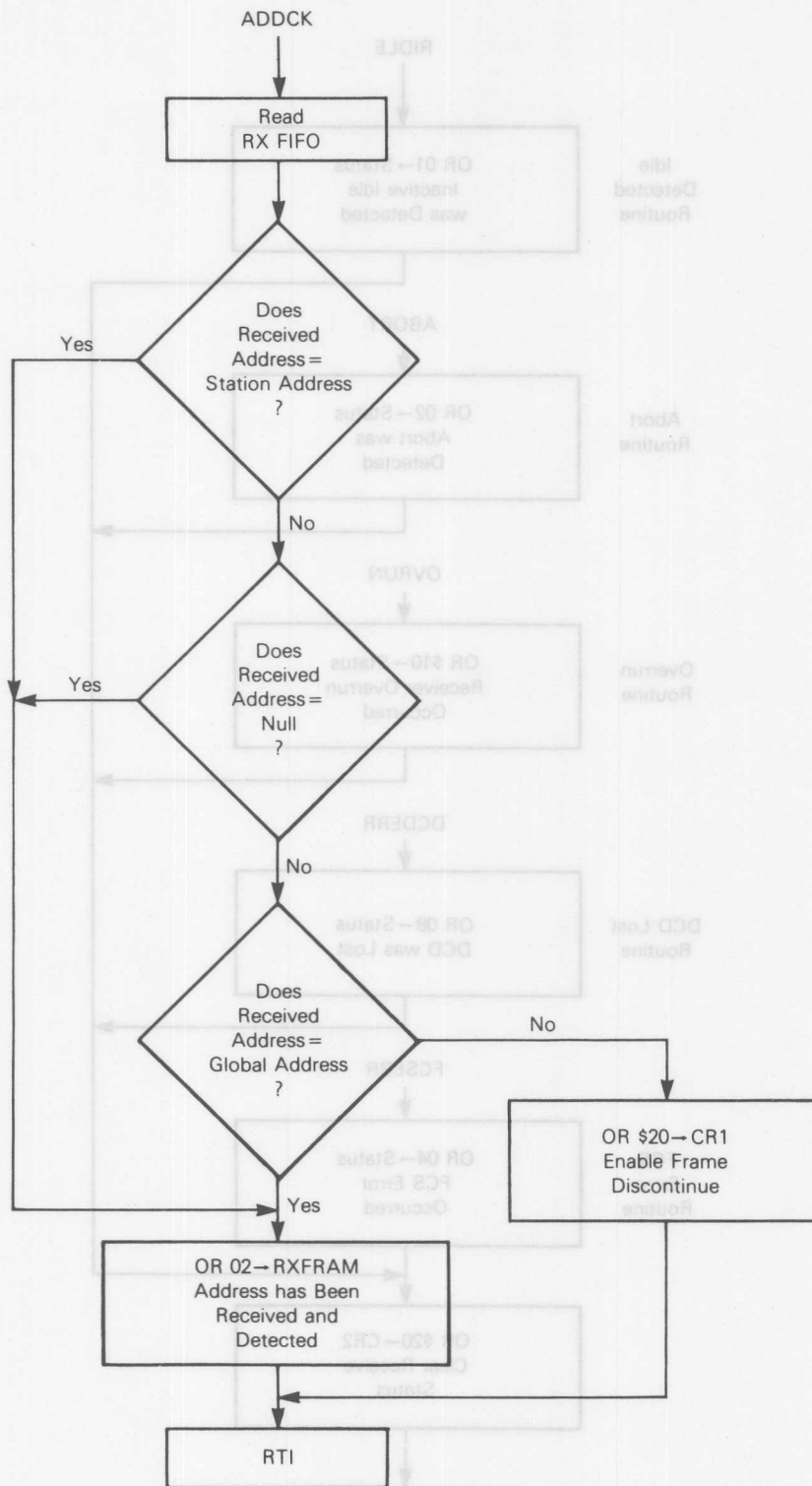


Figure 9. Flowchart of DMA-ADLC Program (Sheet 6 of 9)

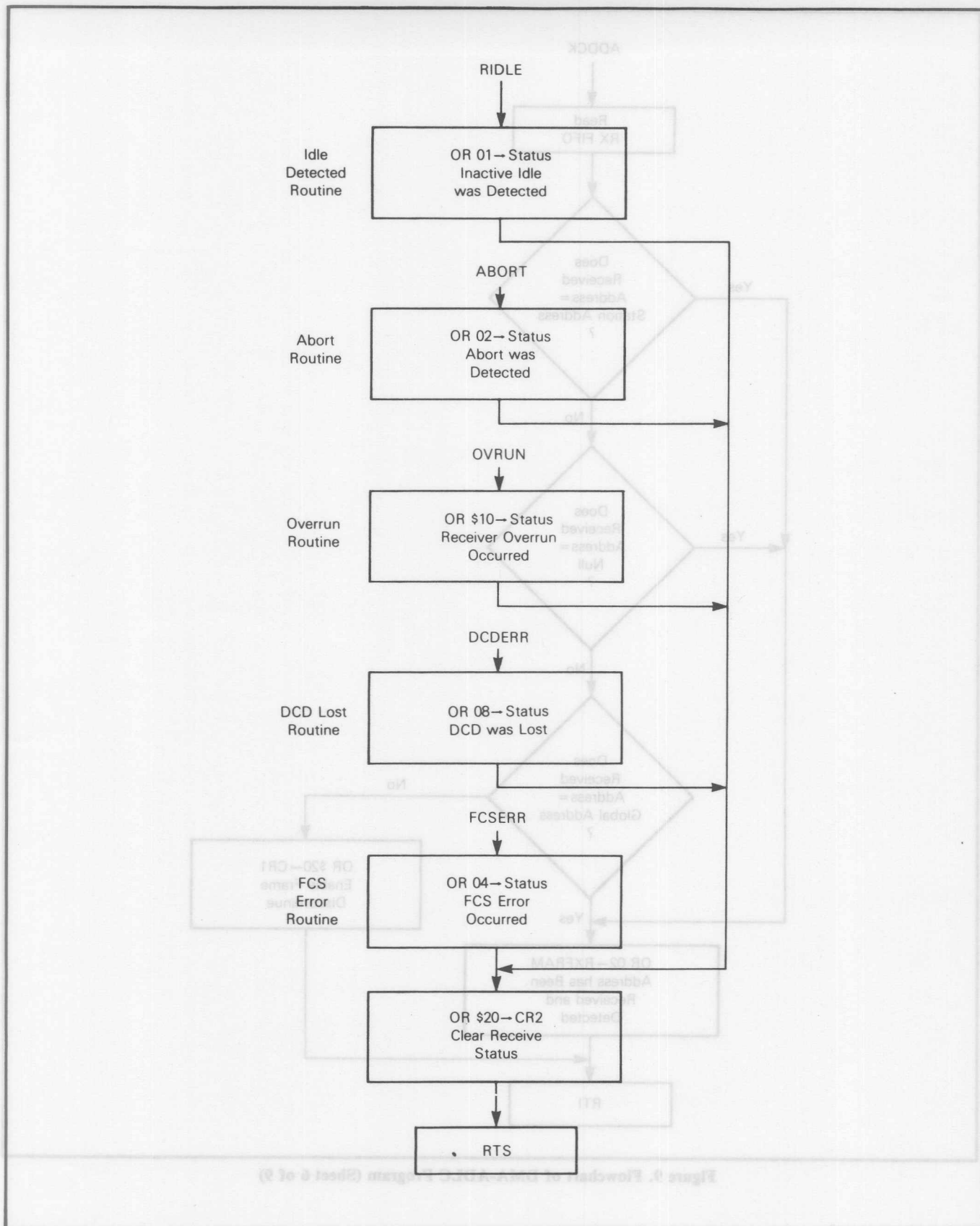


Figure 9. Flowchart of DMA-ADLC Program (Sheet 7 of 9)

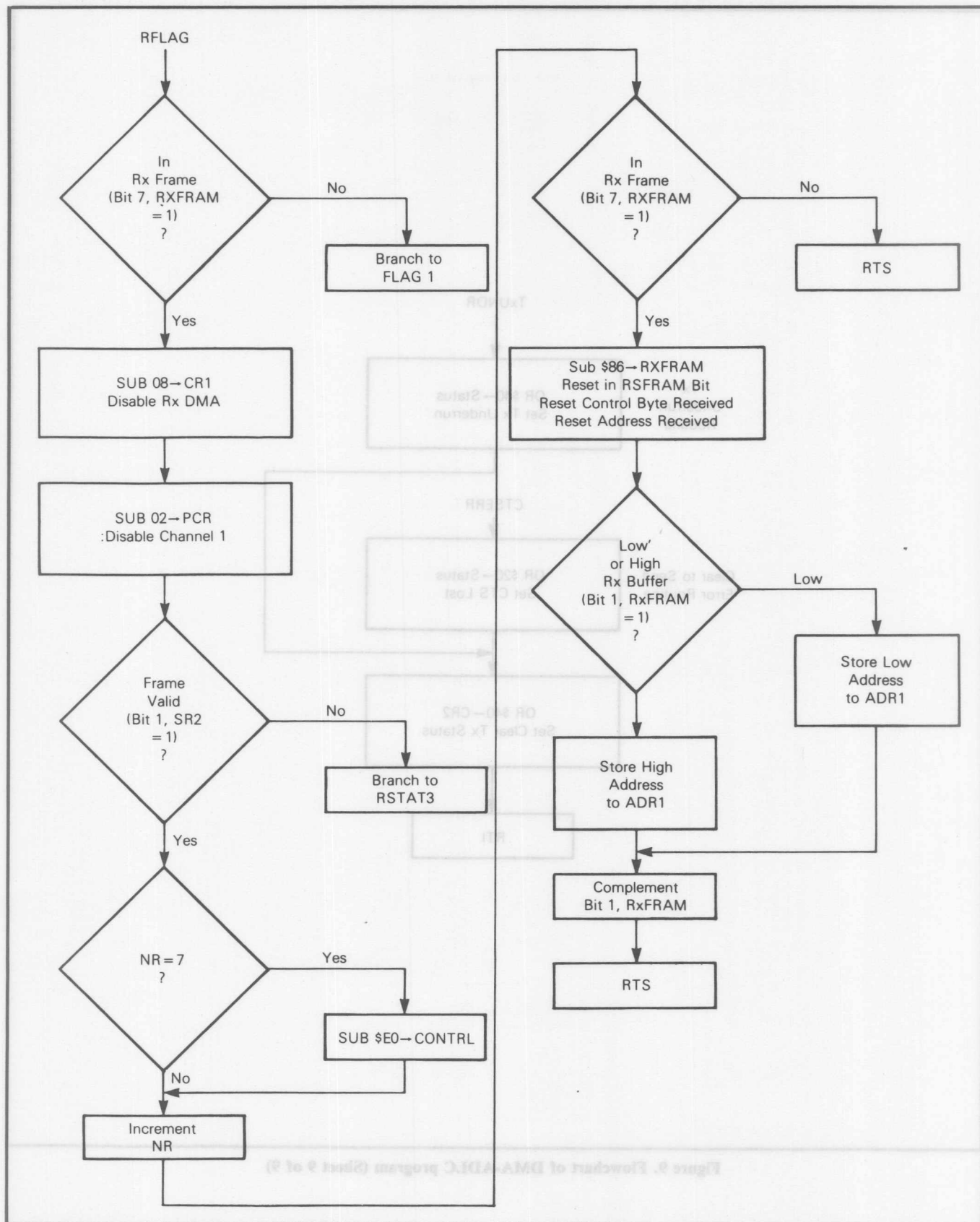


Figure 9. Flowchart of DMA-ADLC Program (Sheet 8 of 9)

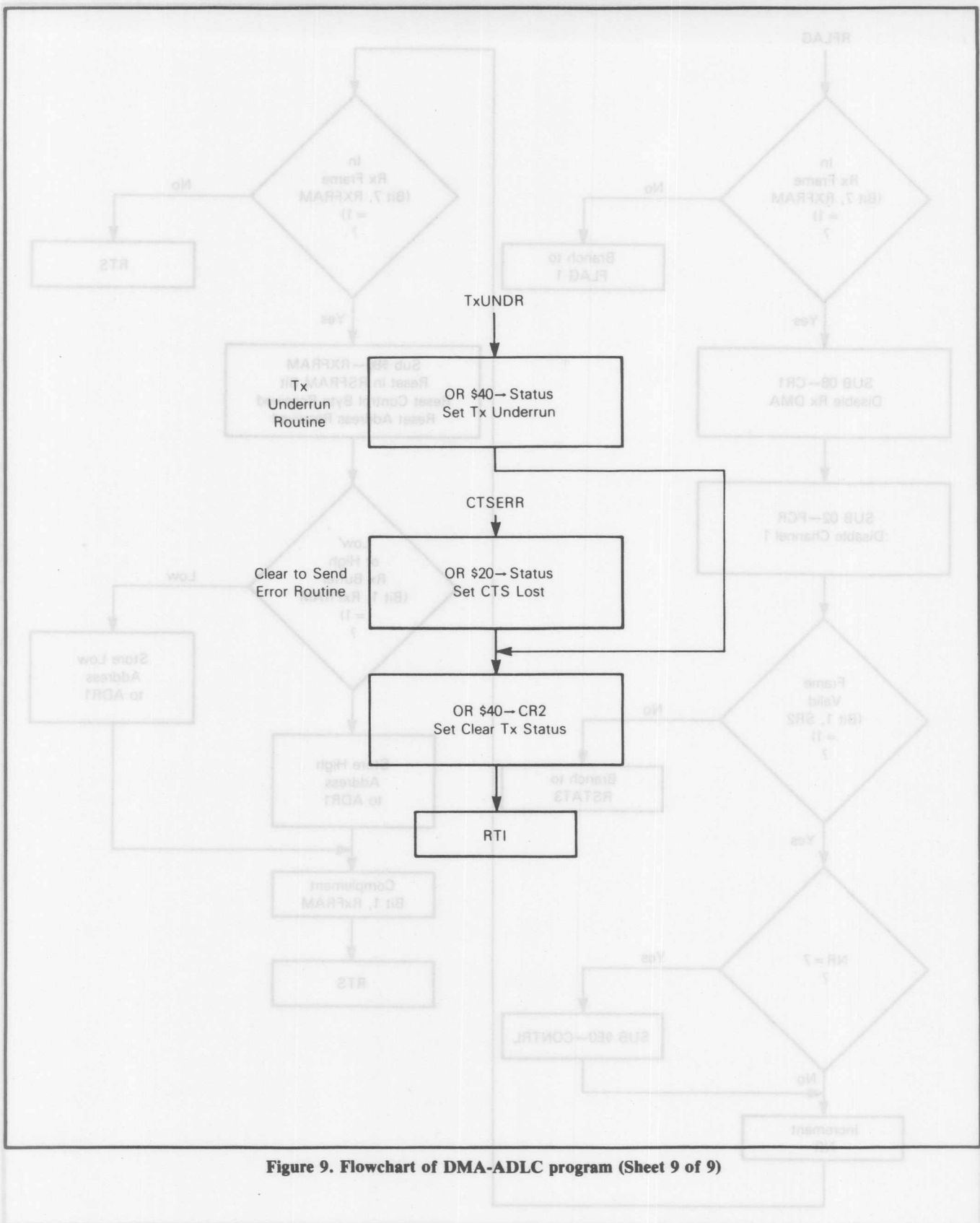


Figure 9. Flowchart of DMA-ADLC program (Sheet 9 of 9)

```

000001      NAM      DMAADLC
000002      * MAY 19,1981
000003      OPT      O,NOG,LLE=82

000005      * THIS PROGRAM IS TO DEMONSTRATE THE MC6854 ADLC
000006      * AS USED WITH THE MC6844 DMA CONTROLLER AND
000007      * THE MC6809 MPU CHIP DESIGNED FOR DMA AND
000008      * DYNAMIC MEMORY REFRESH.

000100      BF40      A ADRG0H EQU      $BF40      DMA ADD REG 0 HIGH ADD
000101      BF41      A ADRG0L EQU      $BF41      DMA REG 0 LOW ADD
000102      BF42      A BCRG0H EQU      $BF42      DMA BYTE COUNT REG 0 HI ADD
000103      BF43      A BCRG0L EQU      $BF43      DMA BYTE COUNT REG 0 LO ADD
000104      BF44      A ADRG1H EQU      $BF44      DMA SDD REG 1 HIGH ADD
000105      BF45      A ADRG1L EQU      $BF45      DMA ADD REG 1 LOW ADD
000106      BF46      A BCRG1H EQU      $BF46      DMA BYTE COUNT REG 1 HI ADD
000107      BF47      A BCRG1L EQU      $BF47      DMA BYTE COUNT REG 1 LO ADD
000108      BF50      A CNTRL0 EQU      $BF50      DMA CHAN 0 CONTROL REG
000109      BF51      A CNTRL1 EQU      $BF51      DMA CHAN 1 CONTROL REG
000110      BF54      A DMAPCR EQU      $BF54      DMA PRIORITY CONTROL REG
000111      BF55      A DMAICR EQU      $BF55      DMA IRQ CONTROL REG
000112      BF56      A DMADCR EQU      $BF56      DMA DATA CHAIN CONTROL REG

000200      BF00      A STATS1 EQU      $BF00      ADLC STATUS #1 ADD
000201      BF01      A STATS2 EQU      $BF01      ADLC STATUS #2 ADDRESS REG.
000202      BF02      A RXFIFO EQU      $BF02      ADLC RXFIFO ADDRESS
000203      BF00      A ADLCR1 EQU      $BF00      ADLC CONTROL REG #1 ADD
000204      BF01      A ADLCR2 EQU      $BF01      ADLC CONTROL REG #2 ADD
000205      BF01      A ADLCR3 EQU      $BF01      ADLC CONTROL REG #3 ADD
000206      BF03      A ADLCR4 EQU      $BF03      ADLC CONTROL REG #4 ADD
000207      BF02      A TXFIFO EQU      $BF02      ADLC TXFIFO ADD

000300      BF10      ORG      $BF10
000301      BF10      00      A STATUS FCB      $00      SOFTWARE CONDITION REGISTER
000302      BF11      00      A TXFRAM FCB      $00      TRANSMIT SOFTWARE STATUS REG
000303      BF12      00      A RXFRAM FCB      $00      RECEIVE SOFTWARE STATUS REG
000304      BF13      4000      A TXBUF1 FDB      $4000      STARTING ADD OF 1ST TX BUFFER
000305      BF15      4200      A TXBUF2 FDB      $4200      STARTING ADD OF 2ND TX BUFFER
000306      BF17      5000      A RXBUF1 FDB      $5000      START ADD OF 1ST RECV BUFF
000307      BF19      5200      A RXBUF2 FDB      $5200      START ADD OF 2ND RECV BUFF
000308      BF1B      AA      A ADRES1 FCB      $AA      STATION ADDRESS
000309      BF1C      00      A ADRES2 FCB      $00      NULL ADDRESS
000310      BF1D      FF      A ADRES3 FCB      $FF      GLOBAL ADDRESS
000311      BF1E      00      A CONTRL FCB      $00      RECEIVED CONTROL WORD STORAGE LO
000312      BF1F      00      A RFMCNT FCB      $00
000313      BF20      00      A TFMCNT FCB      $00
000314      BF21      55      A OUTADD FCB      $55
000315      BF22      00      A OUTCTL FCB      $00
000316      BF23      00      A DMA0IM FCB      $00      DMA CHAN 0 CONTROL REG IMAGE
000317      BF24      00      A DMA1IM FCB      $00      DMA CHAN 1 CONTROL REG IMAGE
000318      BF25      00      A CR1IMG FCB      $00      ADLC CONTROL REG 1 IMAGE
000319      BF26      00      A CR2IMG FCB      $00      ADLC CONTROL REG 2 IMAGE
000320      BF27      00      A CR3IMG FCB      $00      ADLC CONTROL REG 3 IMAGE
000321      BF28      00      A CR4IMG FCB      $00      ADLC CONTROL REG 4 IMAGE

```

Figure 10. Priority Mode Program Listing (Sheet 1 of 11)

20

```

00059A A000      ORG $A000
00060      00BF A SETDP $BF
00061A A000 86 BF A LDA #$BF
00062A A002 1F 8B A TFR A,DP
00063A A004 BE A066 A START LDX HRDINT
00064A A007 BF FFF8 A STX $FFF8
00065A A00A 10CE 1FFF A INIT LDS #$1FFF SET UP STACK
00066A A00E 1A 10 SEI SET IRQ MASK
00067A A010 86 01 A LDAA #$01 SET UP DMA IRQ CON REG
00068A A012 97 55 A STAA DMAICR
00069A A014 97 54 A STAA DMAPCR
00070A A016 9E 13 A LDX TXBUF1 SET UP XMIT ADD CTR IN DMA
00071A A018 9F 40 A STX ADRG0H
00072A A01A 9E 17 A LDX RXBUF1 SET UP RECV ADD CTR IN DMA
00073A A01C 9F 44 A STX ADRG1H
00074A A01E 8E 0200 A LDX #$200 SET UP CHAN 0 BCR (XMIT)
00075A A021 9F 42 A STX BCRG0H WITH 1028 COUNT
00076A A023 86 05 A LDAA #$05 SET UP CHAN 0 CONT REG (XMIT)
00077A A025 97 50 A STAA CNTRL0
00078A A027 86 04 A LDAA #$04 SET UP CHAN 1 CONT REG (RECV)
00079A A029 97 51 A STAA CNTRL1
00080A A02B 86 C1 A LDAA #$C1 ACCESS CR4
00081A A02D 97 00 A STAA ADLCR1
00082A A02F C6 1F A LDAB #$1F SET UP CONTROL REG 4 IN ADLC
00083A A031 D7 03 A STAB ADLCR4 8 BIT WLS, NRZ, FLAG-FLAG
00084A A033 D7 28 A STAB CR4IMG
00085A A035 C6 10 A LDAB #$10 SET UP CONT REG 3 IN ADLC
00086      *ENABLE FLAG DETECT IN RECV
00087A A037 D7 01 A STAB ADLCR3
00088A A039 D7 27 A STAB CR3IMG
00089A A03B 86 C0 A LDAA #$C0 ACCESS CR2
00090A A03D 97 00 A STAA ADLCR1
00091A A03F 97 25 A STAA CR1IMG
00092A A041 C6 05 A LDAB #$05 SET UP ADLC CONT REG 2
00093      *PRIORITY LOGIC ENABLE, 1 BYTE XFER, FLAG IDLE
00094A A043 D7 01 A STAB ADLCR2
00095A A045 D7 26 A STAB CR2IMG SAVE IN IMAGE
00096A A047 17 014C A196 LBSR XMIT ENABLES DMA MODE OF OPERATION
00097A A04A 96 25 A LDAA CR1IMG
00098A A04C 80 C0 A SUBA #$C0 TURN ON XMIT SECTION IN ADLC
00099A A04E 8A 06 A ORAA #$06 ENA IRQ XMIT AND RECV.
00100A A050 97 25 A STAA CR1IMG
00101A A052 97 00 A STAA ADLCR1
00102A A054 1C EF CLI
00103A A056 20 00 A058 BRA WAIT

```

Figure 10. Priority Mode Program Listing (Sheet 3 of 11)

00105 *WAIT IS A LOOP THAT WOULD BE THE NORMAL OPERATIONAL
 00106 *PROGRAM CONTROLLING THE MPU AND OTHER FUNCTIONS
 00107 *OF THE SYSTEM

00110A A058 12 WAIT NOP
 00111A A059 12 NOP
 00112A A05A 96 10 A LDAA STATUS
 00113A A05C 2E 04 A062 BGT SOFT
 00114A A05E 12 NOP
 00115A A05F 12 NOP
 00116A A060 20 F6 A058 BRA WAIT

00118 *SOFT WOULD BE AN AREA WHERE PROBLEMS THAT
 00119 *HAVE OCCURRED SUCH AS A LOSS OF CARRIER (DCD)
 00120 *A RECEIVED ABORT, TX UNDERRUN, OR A LOSS OF CTS
 00121 *WOULD BE HANDLED IN SETTING UP SPECIAL FRAME
 00122 *(SEQUENCED FORMAT) TO INDICATE WHAT IS REQUIRED.

00124A A062 0F 10 A SOFT CLR STATUS
 00125A A064 20 F2 A058 BRA WAIT

Figure 10. Priority Mode Program Listing (Sheet 4 of 11)

00127

*INTERUPTS

00129

*HARDWARE INTERRUPT IS THE AREA OF THE PROGRAM

00130

*THAT SERVICES THE ADLC AND THE DMA ONCE TRANSFERS

00131

*HAVE BEEN STARTED. IF A SYSTEM WOULD NOT USE IRQ

00132

*OR NMI A POLLING ROUTINE WOULD BE NECESSARY

00133

*TO SERVE THIS FUNCTION.

00136A A066 96 50

A

HRDINT

LDAA

CNTRL0

IS IT FROM DMA

00137A A068 2B 05

A06F

BMI

HIRQ2

YES-BRANCH

00138A A06A 96 00

A

LDAA

STATS1

IS IRQ FROM ADLC

00139A A06C 2B 04

A072

BMI

HIRQ1

YES-BRANCH

00140A A06E 3B

RTI

NO-RETURN FROM IRQ

00142

*IF OTHER PERIPHERIALS WERE ENABLED FOR IRQ

00143

*THEY IN TURN WOULD BE POLLED FOR IRQ

00145A A06F 16 00AF

A121

HIRQ2

LBRA

HIRQ02

SAVE ADLC STATUS

00146A A072 97 29

A

HIRQ1

STAA

SRIIMG

KEEP IRQ MASKED

00147A A074 8A 10

A

ORAA

#S10

ACC A TO CCR

00148A A076 1F 8A

TAP

BRANCH IF RXFIFO NEEDS SERVICE

00149A A078 25 68

A0E2

BCS

RDATA

BRANCH IF STATUS REG 2 NEEDS SER

00150A A07A 29 74

A0F0

BVS

RSTAT2

BRANCH IF RXFLAG DETECTED

00151A A07C 2B 22

A0A0

BMI

RFLAG

RELOAD STATUS #1 CONTENTS TO ACC

00152A A07E 96 29

A

HIRQ1A

LDAA

SRIIMG

ROLA

00153A A080 49

ROLA

TXLOAD

00154A A081 2B 07

A08A

BMI

TXLOAD

TRANSMIT DATA REG AVAIL

00155A A083 49

ROLA

TXUNDR

00156A A084 2B 05

A08B

BMI

TXUNDR

CLEAR TO SEND LOST

00157A A086 49

ROLA

TXUNDR

00158A A087 2B 0A

A093

BMI

CTSERR

TXUNDR

00159A A089 3B

RTI

TXUNDR

00160A A08A 3F

TXLOAD

SWI

TXUNDR

00161A A08B 96 10

A

TXUNDR

LDAA

STATUS

TXUNDR

00162

*IN STATUS

00163A A08D 8A 40

A

ORAA

#S40

STATUS

00164A A08F 97 10

A

STAA

STATUS

CLRTXS

00165A A091 20 06

A099

BRA

CLRTXS

STATUS

00166A A093 96 10

A

CTSERR

LDAA

STATUS

STATUS

00167A A095 8A 20

A

ORAA

#S20

STATUS

00168A A097 97 10

A

STAA

STATUS

CR2IMG

00169A A099 96 26

A

CLRTXS

LDAA

CR2IMG

CR2IMG

00170A A09B 8A 40

A

ORAA

#S40

ADLCR2

00171A A09D 97 01

A

STAA

ADLCR2

RTI

00172A A09F 3B

RTI

RXFRAM

00173A A0A0 96 12

A

RFLAG

LDAA

RXFRAM

RXFRAM

00174A A0A2 2A 31

A0D5

BPL

RFLAG1

RFLAG1

00175A A0A4 17 00A8

A14F

LPSR

RDMAOF

RDMAOF

00176A A0A7 D5 26

A

LDAB

CR2IMG

CR2IMG

00177A A0A9 CA 20

A

ORAB

#S20

ADLCR2

00178A A0AB D7 01

A

STAB

ADLCR2

ADLCR2

00179A A0AD 12

NOP

GIVE IT TIME TO DO IT

Figure 10. Priority Mode Program Listing (Sheet 5 of 11)

```

00180A A0AE 96 01 A LDAA STATS2 YES-CHECK IF FRAME VALID
00181A A0B0 97 2A A STAA SR2IMG SAVE IN IMAGE
00182A A0B2 8A 10 A ORAA #$10
00183A A0B4 1F 8A TAP
00184A A0B6 28 42 A0FA BVC RSTAT3 BRANCH IF NOT VALID
00185A A0B8 D6 1E A LDAB CONTRL INC CONTROL NR COUNT
00186A A0BA C4 E0 A ANDB #$E0 CLEAR IF 7 AND INC TO 1
00187A A0BC C1 E0 A CMPB #$E0 IS IT 7 YET
00188A A0BE 27 0F A0CF BEQ RFLAG3 NO-BRANCH
00189A A0C0 D6 1E A LDAB CONTRL YES-CLEAR NR COUNT TO ZERO
00190A A0C2 C0 E0 A SUBB #$E0
00191A A0C4 CB 20 A RFLAG4 ADDB #$20 INC NR COUNT
00192A A0C6 D7 1E A STAB CONTRL
00193A A0C8 17 0181 A24C LBSR GETLST
00194A A0CB 17 00A7 A175 LBSR RXEND GO PREPARE FOR NEXT FRAME
00195A A0CE 3B RFLAG9 RTI
00196A A0CF D6 1E A RFLAG3 LDAB CONTRL
00197A A0D1 C0 E0 A SUBB #$E0
00198A A0D3 20 EF A0C4 BRA RFLAG4

```

```

00200A A0D5 96 12 A RFLAG1 LDAA RXFRAM CHECK IF 1ST FLG BIT SET
00201A A0D7 8A 08 A ORAA #$08
00202A A0D9 97 12 A STAA RXFRAM
00203A A0DB 96 26 A LDAA CR2IMG
00204A A0DD 8A 20 A ORAA #$20 CLEAR RX STATUS
00205A A0DF 97 01 A STAA ADLCR2
00206A A0E1 3B RTI
00207A A0E2 D6 02 A RDATA LDAB RXFIFO GO GET AVAILABLE DATA
00208A A0E4 96 12 A LDAA RXFRAM
00209A A0E6 D7 2B A STAB INCRTL SAVE CONTROL BYTE
00210A A0E8 8A 84 A ORAA #$84 DECLARE INFRAME STATUS
00211A A0EA 97 12 A STAA RXFRAM
00212A A0EC 17 00E6 A1D5 LBSR RDMAON TURN ON RECV DMA MODE
00213A A0EF 3B RDATA9 RTI

```

```

00215 *READS STATUS REG 2 OF ADLC AND CHECKS FOR ERRORS
00216 *OR IF RECEIVED FRAME WAS VALID.

```

```

00218A A0F0 96 01 A RSTAT2 LDAA STATS2 GETS STATUS 2 FROM ADLC
00219A A0F2 97 2A A STAA SR2IMG
00220A A0F4 8A 10 A ORAA #$10
00221A A0F6 1F 8A TAP
00222A A0F8 25 10 A10A RCS ADDCK BRANCH IF ADDRESS PRESENT
00223A A0FA 27 11 A10D RSTAT3 BEQ RIDLE BRANCH IF IDLE DETECTED
00224A A0FC 2B 13 A111 BMI ABORT BRANCH IF ABORT DETECTED
00225A A0FE 96 2A A LDAA SR2IMG
00226A A100 49 ROLA
00227A A101 2B 1A A11D BMI OVRUN RECEIVER OVERRUN ERROR
00228A A103 49 ROLA
00229A A104 2B 13 A119 BMI DCDERR DATA CARRIER LOST
00230A A106 49 ROLA
00231A A107 2B 0C A115 BMI PCSERR FRAME CHECK SEQUENCE ERROR
00232A A109 3B RTI
00233A A10A 8D 23 A12F ADDCK BSR CKADD SEE IF THIS IS OUR ADDRESS
00234A A10C 3B RTI

```

Figure 10. Priority Mode Program Listing (Sheet 6 of 11)

```

00235A A10D 17 0103 A213 RIDLE LBSR IDLE INDICATE THAT AN INACTIVE
00236 *IDLE WAS DETECTED
00237A A110 3B RTI

```

```

00239A A111 17 0110 A224 ABORT LBSR RABORT INDICATE AN ABORT WAS RECEIVED
00240A A114 3B RTI
00241A A115 17 0116 A22E FCSERR LBSR CRCERR INDICATE FCS ERROR OCCURED
00242A A118 3B RTI
00243A A119 17 011C A238 DCDERR LBSR DCDLST DATA CARRIER DETECT FROM
00244 *MODEM WAS LOST
00245A A11C 3B RTI
00246A A11D 17 0122 A242 OVRUN LBSR OVRUN1 SET BIT TO INDICATE OVERRUN
00247A A120 3B RTI

```

```

00249 *DMA SERVICE INTERUPT

```

```

00251A A121 96 26 A HIRQ02 LDAA CR2IMG SET LAST DATA BIT IN CR2
00252A A123 8A 10 A ORAA #$10 (AUTO RESET)
00253A A125 97 01 A STAA ADLCR2
00254A A127 8D 39 A162 BSR TDMAOF TURN OFF DMA MODE
00255A A129 17 00C1 A1ED LBSR TDMAON LOAD DMA ADDRESS REG AND BCR
00256A A12C 8D 68 A196 BSR XMIT
00257A A12E 3B RTI
00258 *
00259 *
00260 *
00261 *
00262 *
00263 *
00264 *
00265 *
00266 *
00267 *THIS ROUTINE UNLOADS THE ADDRESS FROM THE ADLC
00268 *RXFIFO AND COMPARES IT THE STATION ADDRESSES.
00269 *IF CORRECT IT SETS THE ADD. RECV. BIT IN THE
00270 *RXFRAME. IF NOT THIS STATIONS' ADDRESS, CLEAR SYNC IS
00271 *AND THE RECEIVER BEGINS LOOKING FOR THE FLAG CONDITON
00272 *AGAIN TO SYNC ON.

```

```

00274A A12F 34 04 CKADD PSHR
00275A A131 D6 02 A LDAB RXFIFO GET ADD BYTE
00276A A133 D1 1B A CMPB ADRES1 COMPARE RECV DATA TO POSSIBLE
00277A A135 27 0F A146 BEQ CKADD2 STATION ADDRESSES
00278A A137 D1 1C A CMPB ADRES2
00279A A139 27 0B A146 BEQ CKADD2 YES-BRANCH
00280A A13B D1 1D A CMPB ADRES3
00281A A13D 27 07 A146 BEQ CKADD2 YES-BRANCH
00282 * NO ADDRESS MATCH THEN CLEAR RECEIVE SYNC
00283A A13F D6 25 A LDAB CR1IMG
00284A A141 CA 20 A ORAB #$20 CLEAR SYNC IN ADLC
00285A A143 D7 00 A STAB ADLCR1 DO IT
00286A A145 39 RTS
00287A A146 D6 12 A CKADD2 LDAB RXFRAM
00288A A148 CA 02 A ORAB #$02

```

Figure 10. Priority Mode Program Listing (Sheet 7 of 11)

PAGE 008 DMAADLC .SA:1 DMAADL

```
00289A A14A D7 12 A STAB RXFRAM SET ADD BIT IN RXFRAM
00290A A14C 35 04 CKADD9 PULB
00291A A14E 39 RTS
```

```
00293 *THIS SUBROUTINE TURNS OFF DMA CHAN 1 ENABLE AND
00294 *ADLC RECEIVE MODE OF OPERATION.
00295A A14F 34 02 RDMAOF PSHA
00296A A151 96 25 A LDAA CR1IMG GET IMAGE OF CR1
00297A A153 80 08 A SUBA #$08 DISABLE RX DMA MODE IN ADLC
00298A A155 97 25 A STAA CR1IMG
00299A A157 97 00 A STAA ADLCR1
00300A A159 96 54 A LDAA DMAPCR FETCH DMA PCR DATA
00301A A15B 80 02 A SUBA #$02 RESET CHAN 1 ENABLE BIT
00302A A15D 97 54 A STAA DMAPCR
00303A A15F 35 02 PULA
00304A A161 39 RTS
```

```
00306 *TURNS OFF TX DMA MODE IN ADLC AND DMA CHAN #0
00307 *IS DISABLED
00308A A162 34 02 TDMAOF PSHA
00309A A164 96 25 A LDAA CR1IMG
00310A A166 80 10 A SUBA #$10 RESET TXDMA BIT IN CR1
00311A A168 97 25 A STAA CR1IMG
00312A A16A 97 00 A STAA ADLCR1 DO IT
00313A A16C 96 54 A LDAA DMAPCR GET PCR CONTENTS
00314A A16E 80 01 A SUBA #$01 RESET CHAN #0 ENABLE BIT
00315A A170 97 54 A STAA DMAPCR DO IT
00316A A172 35 02 PULA
00317A A174 39 RTS
```

```
00319 *THIS ROUTINE LOADS THE ALTERNATE RXBUFFER ADDRESS
00320 *INTO THE DMA, CLEARS THE IN FRAME BIT, AND SETS
00321 *THE POINTER TO THE NEXT RXBUFFER AREA TO BE LOADED
00322 * INTO THE DMA
```

```
00324A A175 34 02 RXEND PSHA
00325A A177 96 12 A LDAA RXFRAM
00326A A179 85 80 A BITA #$80 TEST IF IN FRAME
00327A A17B 27 16 A193 BEQ RXEND9 NO-BRANCH-LEAVE ROUTINE
00328A A17D 80 86 A SUBA #$86 YES-CLEAR IN FRAME BIT &ADD & CO
00329A A17F 85 01 A BITA #$01 TEST HI OR LO ADDRESS NEXT
00330A A181 27 08 A18B BEQ RXEND1 BRANCH TO LOAD LOW ADD
00331A A183 80 01 A SUBA #$01 RESET ADD START BIT IN RXFRAM
00332A A185 9E 17 A LDXX RXBUF1 LOAD HIGH ADDRESS
00333A A187 9F 44 A STX ADRG1H
00334A A189 20 06 A191 BRA RXEND2
00335A A18B 8B 01 A RXEND1 ADDA #$01
00336A A18D 9E 19 A LDXX RXBUF2 LOAD LOW ADDRESS
00337A A18F 9F 44 A STX ADRG1H
00338A A191 97 12 A RXEND2 STAA RXFRAM
00339A A193 35 02 RXEND9 PULA
00340A A195 39 RTS
```

Figure 10. Priority Mode Program Listing (Sheet 8 of 11)

```

00342      *SUBROUTINE TO LOAD THE TXFIFO WITH THE ADDRESS
00343      *AND CONTROL WORDS, AND TO ENABLE THE ADLC IN
00344      *THE TX DMA MODE OF OPERATION.(XMIT SECTION OF THE
00345      *ADLC IS ONCE THE INITIAL SEQUENCE HAS BEEN PERFORMED.)

```

```

00347A A196 34 02      XMIT  PSHA
00348A A198 34 04      PSHB
00349A A19A D6 11      A      LDAB      TXFRAM      DETERMINE WHICH TXBUF TO USE
00350A A19C C5 01      A      BITB      #$01      IS IT #1
00351A A19E 27 06      A1A6    BEQ      XMIT2      YES--BRANCH
00352A A1A0 9E 15      A      LDX      TXBUF2     NO---IT'S #2
00353A A1A2 C0 01      A      SUBB      #$01      CHANGE FOR NEXT TIME
00354A A1A4 20 04      A1AA    BRA      XMIT3
00355A A1A6 9E 13      A XMIT2  LDX      TXBUF1
00356A A1A8 CA 01      A      ORAB      #$01      CHANGE FOR NEXT TIME
00357A A1AA D7 11      A XMIT3  STAB      TXFRAM
00358A A1AC 96 1B      A      LDAA      ADRES1     ADDRESS BYTE SET UP
00359A A1AE A7 84      A      STAA      0,X
00360A A1B0 30 01      INX
00361A A1B2 96 1E      A      LDAA      CONTRL     CONTROL WORD SET UP
00362A A1B4 84 0E      A      ANDA      #$0E      TEST IF 7 FRAMES SENT
00363A A1B6 81 0E      A      CMPA      #$0E
00364A A1B8 27 15      A1CF    BEQ      XMTCLR     YES-BRANCH
00365A A1BA 96 1E      A      LDAA      CONTRL     NO-CONTINUE
00366A A1BC 8B 02      A XMIT1  ADDA      #$02      INCREMENT THE NS COUNT
00367A A1BE A7 84      A      STAA      0,X      LOAD IT OUT
00368A A1C0 97 1E      A      STAA      CONTRL     SAVE THE NEW CONTROL WORD
00369A A1C2 96 25      A      LDAA      CRLIMG
00370A A1C4 8A 10      A      ORAA      #$10      ENABLE DMA MODE OF OPERATION
00371A A1C6 97 00      A      STAA      ADLCR1     DO IT
00372A A1C8 97 25      A      STAA      CRLIMG
00373A A1CA 35 04      PULB
00374A A1CC 35 02      PULA
00375A A1CE 39      RTS
00376A A1CF 96 1E      A XMTCLR  LDAA      CONTRL
00377A A1D1 80 0E      A      SUBA      #$0E      CLR NS FRAME COUNT
00378A A1D3 20 E7      A1BC    BRA      XMIT1

```

```

00380      *ROUTINE TO TURN RECEIVER OPERATIONS OVER TO
00381      *THE DMA CONTROLLER.

```

```

00383A A1D5 34 02      RDMAON  PSHA
00384A A1D7 8E FFFF     A      LDX      $FFFF     SET BYTE COUNT REG TO A SIZE
00385A A1DA 9F 45      A      STX      BCRG1H    LARGER THAN THE EXPECTED FRAME

```

```

00387      *THE ADDRESS REGISTER IS ALREADY SET BUT ITS SET UP
00388      *COULD BE LOCATED HERE ALSO.(IN RXEND SUBROUTINE)

```

```

00390A A1DC 96 54      A      LDAA      DMAPCR     TURN ON DMA CHAN 1 (RCV)
00391A A1DE 8A 02      A      ORAA      #$02
00392A A1E0 97 54      A      STAA      DMAPCR
00393A A1E2 96 25      A      LDAA      CRLIMG     TURN ON ADLC TO DMA MODE
00394A A1E4 8A 08      A      ORAA      #$08     OF OPERATION

```

Figure 10. Priority Mode Program Listing (Sheet 9 of 11)

```

00395A A1E6 97 25 A STAA CR1IMG
00396A A1E8 97 00 A STAA ADLCR1 DO TI
00397A A1EA 35 02 PULA
00398A A1EC 39 RTS

```

```

00400 *SUBROUTINE TO LOAD THE ALTERNATE TX BUFFER
00401 *ADDRESS INTO THE ADDRESS REG OF THE DMA CHAN 0

```

```

00403A A1ED 34 02 TDMAON PSHA
00404A A1EF 96 11 A LDAA TXFRAM GET TX FRAME STATUS
00405A A1F1 85 01 A BITA #$01 TEST WHICH BUFFER TO USE
00406A A1F3 26 08 A1FD BNE TDMON1 BRANCH NOT SET
00407A A1F5 8A 01 A ORAA #$01
00408A A1F7 9E 15 A LDX TXBUF2 SELECT TX BUFFER #2
00409A A1F9 9F 40 A STX ADRG0H SET UP ADD REG IN DMA
00410A A1FB 20 06 A203 BRA TDMON2
00411A A1FD 80 01 A TDMON1 SUBA #$01
00412A A1FF 9E 13 A LDX TXBUF1 SELECT TX BUFFER #1
00413A A201 9F 40 A STX ADRG0H SET UP ADD REG IN DMA
00414A A203 97 11 A TDMON2 STAA TXFRAM
00415A A205 8E 0400 A LDX #$0400 SET UP BCR IN DMA
00416A A208 9F 42 A STX BCRG0H DO IT
00417A A20A 96 54 A LDAA DMAPCR ENABLE CHAN 0 IN DMA
00418A A20C 8A 01 A ORAA #$01
00419A A20E 97 54 A STAA DMAPCR
00420A A210 35 02 PULA
00421A A212 39 RTS

```

```

00423 *SUBROUTINE TO SET THE INACTIVE IDLE BIT IN THE
00424 *STATUS SOFTWARE REGISTER

```

```

00425A A213 34 02 IDLE PSHA
00427A A215 96 10 A LDAA STATUS
00428A A217 8A 01 A ORAA #$01 SET INACTIVE IDLE BIT
00429A A219 97 10 A STAA STATUS
00430A A21B 96 26 A CLEAR LDAA CR2IMG CLEAR RECEIVER STATUS
00431A A21D 8A 20 A ORAA #$20
00432A A21F 97 01 A STAA ADLCR2
00433A A221 35 02 PULA
00434A A223 39 RTS

```

```

00436 *SUBROUTINE TO SET THE ABORT BIT IN THE STATUS
00437 *SOFTWARE REGISTER

```

```

00439A A224 34 02 RABORT PSHA
00440A A226 96 10 A LDAA STATUS SET ABORT BIT
00441A A228 8A 02 A ORAA #$02
00442A A22A 97 10 A STAA STATUS
00443A A22C 20 ED A21B BRA CLEAR
00444 *SUBROUTINE TO SET THE FCS ERROR BIT IN THE
00445 *STATUS SOFTWARE REGISTER

```

Figure 10. Priority Mode Program Listing (Sheet 10 of 11)

```

00447A A22E 34 02          CRCERR PSHA
00448A A230 96 10          A      LDAA     STATUS  SET FCS ERROR BIT
00449A A232 8A 04          A      ORAA     #$04
00450A A234 97 10          A      STAA     STATUS
00451A A236 20 E3 A21B     BRA      CLEAR

```

```

00453          *SUBROUTINE TO SET THE DCD ERROR BIT IN THE
00454          *STATUS SOFTWARE REGISTER

```

```

00456A A238 34 02          DCDLST PSHA
00457A A23A 96 10          A      LDAA     STATUS
00458A A23C 8A 08          A      ORAA     #$08
00459A A23E 97 10          A      STAA     STATUS
00460A A240 20 D9 A21B     BRA      CLEAR

```

```

00462          *SUBROUTINE TO SET RECEIVE OVERRUN BIT IN STATUS
00463          *SOFTWARE REGISTER AND CLEAR THE RECEIVER STATUS

```

```

00465A A242 34 02          OVRUN1 PSHA
00466A A244 96 10          A      LDAA     STATUS  SET RX OVERRUN BIT IN STATUS
00467A A246 8A 10          A      ORAA     #$10
00468A A248 97 10          A      STAA     STATUS
00469A A24A 20 CF A21B     BRA      CLEAR

```

```

00471          * THIS SUBROUTINE TAKES THE LAST BYTE OF DATA
00472          * OUT OF THE RECEIVE FIFO AT THE END OF EACH
00473          * FRAME.

```

```

00475A A24C 34 04          GETLST PSHB
00476A A24E 34 02          PSHA
00477A A250 96 26          A      LDAA     CR2IMG  CLEAR RECEIVE STATUS
00478A A252 8A 20          A      ORAA     #$20    TO ENABLE RDA
00479A A254 97 01          A      STAA     ADLCR2  TO BE READ
00480A A256 12             NOP                GIVE IT TIME TO DO IT
00481A A257 96 00          A      GTLST2 LDAA     STATS1 CHECK FOR DATA
00482A A259 85 01          A      BITA     #$01
00483A A25B 27 08 A265     BEQ      GTLST9  NO DATA---BRANCH
00484A A25D D6 02          A      LDAB     RXFIFO  GET THE DATA BYTE
00485A A25F 9E 44          A      LDX      ADRG1H  GET NEXT ADD OF RX BUFFER
00486A A261 E7 84          A      STAB     0,X
00487A A263 30 01          INX
00488A A265 35 02          GTLST9 PULA
00489A A267 35 04          PULB
00490A A269 39            RTS
00491          END

```

```

TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

```

Figure 10. Priority Mode Program Listing (Sheet 11 of 11)

```

00001      NAM      NOPRI
00002      * MAY 19, 1981
00003      OPT      O,NOG,LLE=82

00005      *NON PRIORITY MODE OF OPERATION WITH DMA AND ADLC
00006      *

00008      * THIS PROGRAM IS TO DEMONSTRATE THE MC6854 ADLC
00009      * AS USED WITH THE MC6844 DMA CONTROLLER AND
00010      * THE MC6809 MPU CHIP DESIGNED FOR DMA AND
00011      * DYNAMIC MEMORY REFRESH.

00013      BF40      A ADRG0H EQU      $BF40      DMA ADD REG 0 HIGH ADD
00014      BF41      A ADRG0L EQU      $BF41      DMA REG 0 LOW ADD
00015      BF42      A BCRG0H EQU      $BF42      DMA BYTE COUNT REG 0 HI ADD
00016      BF43      A BCRG0L EQU      $BF43      DMA BYTE COUNT REG 0 LO ADD
00017      BF44      A ADRG1H EQU      $BF44      DMA SDD REG 1 HIGH ADD
00018      BF45      A ADRG1L EQU      $BF45      DMA ADD REG 1 LOW ADD
00019      BF46      A BCRG1H EQU      $BF46      DMA BYTE COUNT REG 1 HI ADD
00020      BF47      A BCRG1L EQU      $BF47      DMA BYTE COUNT REG 1 LO ADD
00021      BF50      A CNTRL0 EQU      $BF50      DMA CHAN 0 CONTROL REG
00022      BF51      A CNTRL1 EQU      $BF51      DMA CHAN 1 CONTROL REG
00023      BF54      A DMAPCR EQU      $BF54      DMA PRIORITY CONTROL REG
00024      BF55      A DMAICR EQU      $BF55      DMA IRQ CONTROL REG
00025      BF56      A DMADCR EQU      $BF56      DMA DATA CHAIN CONTROL REG

00027      BF00      A STATS1 EQU      $BF00      ADLC STATUS #1 ADD
00028      BF01      A STATS2 EQU      $BF01      ADLC STATUS #2 ADDRESS REG.
00029      BF02      A RXFIFO EQU      $BF02      ADLC RXFIFO ADDRESS
00030      BF00      A ADLCR1 EQU      $BF00      ADLC CONTROL REG #1 ADD
00031      BF01      A ADLCR2 EQU      $BF01      ADLC CONTROL REG #2 ADD
00032      BF01      A ADLCR3 EQU      $BF01      ADLC CONTROL REG #3 ADD
00033      BF03      A ADLCR4 EQU      $BF03      ADLC CONTROL REG #4 ADD
00034      BF02      A TXFIFO EQU      $BF02      ADLC TXFIFO ADD

00036A BF10      ORG      $BF10

00038A BF10      00      A STATUS FCB      $00      SOFTWARE CONDITION REGISTER
00039A BF11      00      A TXFRAM FCB      $00      TRANSMIT SOFTWARE STATUS REG
00040A BF12      00      A RXFRAM FCB      $00      RECEIVE SOFTWARE STATUS REG
00041A BF13      4000      A TXBUF1 FDB      $4000      STARTING ADD OF 1ST TX BUFFER
00042A BF15      4200      A TXBUF2 FDB      $4200      STARTING ADD OF 2ND TX BUFFER
00043A BF17      5000      A RXBUF1 FDB      $5000      START ADD OF 1ST RECV BUFF
00044A BF19      5200      A RXBUF2 FDB      $5200      START ADD OF 2ND RECV BUFF
00045A BF1B      AA      A ADRES1 FCB      $AA      STATION ADDRESS
00046A BF1C      00      A ADRES2 FCB      $00      NULL ADDRESS
00047A BF1D      FF      A ADRES3 FCB      $FF      GLOBAL ADDRESS
00048A BF1E      00      A CONTRL FCB      $00      RECEIVED CONTROL WORD STORAGE LO
00049A BF1F      00      A RFMCNT FCB      $00
00050A BF20      00      A TFMCNT FCB      $00
00051A BF21      55      A OUTADD FCB      $55
00052A BF22      00      A OUTCTL FCB      $00

```

Figure 11. Non-Priority Mode Program Listing (Sheet 1 of 12)


```

00066A A000      ORG      $A000
00067      00BF      A      SETDP    $BF
00068A A000 86      BF      A      LDA      #BF
00069A A002 1F      8B      A      TFR      A,DP
00070A A004 BE      A065      A START  LDX      HRDINT
00071A A007 BF      FFF8      A      STX      $FFF8
00072A A00A 10CE 3FFF      A INIT  LDS      #$3FFF      SET UP STACK
00073A A00E 1A      10      A      SEI      SET IRQ MASK
00074A A010 86      01      A      LDAA     #$01      SET UP DMA IRQ CON REG
00075A A012 97      55      A      STAA     DMAICR
00076A A014 97      54      A      STAA     DMAPCR
00077A A016 9E      13      A      LDX      TXBUF1      SET UP XMIT ADD CTR IN DMA
00078A A018 9F      40      A      STX      ADRG0H
00079A A01A 9E      17      A      LDX      RXBUF1      SET UP RECV ADD CTR IN DMA
00080A A01C 9F      44      A      STX      ADRG1H
00081A A01E 8E      0200      A      LDX      #$200      SET UP CHAN 0 BCR (XMIT)
00082A A021 9F      42      A      STX      BCRG0H      WITH 1024 COUNT
00083A A023 86      05      A      LDAA     #$05      SET UP CHAN 0 CONT REG (XMIT)
00084A A025 97      50      A      STAA     CNTRL0
00085A A027 86      04      A      LDAA     #$04      SET UP CHAN 1 CONT REG (RECV)
00086A A029 97      51      A      STAA     CNTRL1
00087A A02B 86      C1      A      LDAA     #$C1      ACCESS CR4
00088A A02D 97      00      A      STAA     ADLCR1
00089A A02F C6      1F      A      LDAB     #$1F      SET UP CONTROL REG 4 IN ADLC
00090A A031 D7      03      A      STAB     ADLCR4      8 BIT WLS, NRZ, FLAG-FLAG
00091A A033 D7      28      A      STAB     CR4IMG
00092A A035 5F      CLR      SET UP CONT REG 3 IN ADLC
00093      *FLAG DETECT NOT ENABLED
00094A A036 D7      01      A      STAB     ADLCR3
00095A A038 D7      27      A      STAB     CR3IMG
00096A A03A 86      C0      A      LDAA     #$C0      ACCESS CR2
00097A A03C 97      00      A      STAA     ADLCR1
00098A A03E 97      25      A      STAA     CR1IMG
00099A A040 C6      04      A      LDAB     #$04      SET UP ADLC CONT REG 2
00100      * 1 BYTE TRANSFER, FLAG IDLE
00101A A042 D7      01      A      STAB     ADLCR2
00102A A044 D7      26      A      STAB     CR2IMG      SAVE IN IMAGE
00103A A046 17      0190      AID9      LBSR     XMIT      ENABLES DMA MODE OF OPERATION
00104A A049 96      25      A      LDAA     CR1IMG
00105A A04B 80      C0      A      SUBA     #$C0      TURN ON XMIT SECTION IN ADLC
00106A A04D 8A      06      A      ORAA     #$06      ENA IRQ XMIT AND RECV.
00107A A04F 97      25      A      STAA     CR1IMG
00108A A051 97      00      A      STAA     ADLCR1
00109A A053 1C      EF      CLI
00110A A055 20      00      A057      BRA      WAIT

```

Figure 11. Non-Priority Mode Program Listing (Sheet 3 of 12)

00112 *WAIT IS A LOOP THAT WOULD BE THE NORMAL OPERATIOAL
 00113 *PROGRAM CONTROLLING THE MPU AND OTHER FUNCTIONS
 00114 *OF THE SYSTEM

00117A	A057	12			WAIT	NOP	
00118A	A058	12				NOP	
00119A	A059	96	10	A		LDAA	STATUS
00120A	A05B	2E	04	A051		BGT	SOFT
00121A	A05D	12				NOP	
00122A	A05E	12				NOP	
00123A	A05F	20	F6	A057		BRA	WAIT

00125 *SOFT WOULD BE AN AREA WHERE PROBLEMS THAT
 00126 *HAVE OCCURRED SUCH AS A LOSS OF CARRIER (DCD)
 00127 *A RECEIVED ABORT, TX UNDERRUN, OR A LOSS OF CTS
 00128 *WOULD BE HANDLED IN SETTING UP SPECIAL FRAME
 00129 *(SEQUENCED FORMAT) TO INDICATE WHAT IS REQUIRED.

00131A	A061	0F	10	A	SOFT	CLR	STATUS
00132A	A063	20	F2	A057		BRA	WAIT

Figure 11. Non-Priority Mode Program Listing (Sheet 4 of 12)

00134 *INTERUPTS

00136 *HARDWARE INTERRUPT IS THE AREA OF THE PROGRAM
 00137 *THAT SERVICES THE ADLC AND THE DMA ONCE TRANSFERS
 00138 *HAVE BEEN STARTED. IF A SYSTEM WOULD NOT USE IRQ
 00139 *OR NMI A POLLING ROUTINE WOULD BE NECESSARY
 00140 *TO SERVE THIS FUNCTION. POLLING HOWEVER WOULD
 00141 *GREATLY RESTRICT THE MPU DURING TRANSFERS.
 00142

00145A A065 96 50 A HRDINT LDAA CNTRL0 IS IT FROM DMA
 00146A A067 2B 05 A06E BMI HIRQ2 YES-BRANCH
 00147A A069 96 00 A LDAA STATS1 IS IRQ FROM ADLC
 00148A A06B 2B 06 A073 BMI HIRQ1 YES-BRANCH
 00149A A06D 3B RTI NO-RETURN FROM IRQ

00151 *IF OTHER PERIPHERIALS WERE ENABLED FOR IRQ
 00152 *THEY IN TURN WOULD BE POLLED FOR IRQ

00154A A06E 16 00DD A14E HIRQ2 LBRA HIRQ02
 00155A A071 97 2C A STAA SRLIM2 SAVE FOR SAFETY BEFORE CLR
 00156A A073 97 29 A HIRQ1 STAA SRLIMG SAVE ADLC STATUS
 00157A A075 96 29 A HIRQ1A LDAA SRLIMG RELOAD STATUS #1 CONTENTS TO ACC
 00158A A077 49 ROLA
 00159A A078 49 ROLA
 00160A A079 2B 0F A08A BMI TXUNDR TRANSMITTER UNDERFLOW
 00161A A07B 49 ROLA
 00162A A07C 2B 18 A095 BMI CTSERR CLEAR TO SEND LOST
 00163A A07E 96 29 A HIRQ3 LDAA SRLIMG
 00164A A080 8A 10 A ORAA #\$10 KEEP IRQ MASKED
 00165A A082 1F 8A TAP ACC A TO CCR
 00166A A084 29 1E A0A4 BVS RSTAT2 BRANCH IF SR2 NEEDS SERVICE
 00167A A086 16 01CD A255 LBRA CLEAR
 00168A A089 3F TXLOAD SWI NEVER SUPPOSED TO BE HERE
 00169A A08A 96 10 A TXUNDR LDAA STATUS SET BIT OF TX UNDERRUN
 00170 *IN STATUS
 00171A A08C 8A 40 A ORAA #\$40
 00172A A08E 97 10 A STAA STATUS
 00173A A090 96 29 A LDAA SRLIMG
 00174A A092 80 20 A SUBA #\$20
 00175A A094 20 DD A073 BRA HIRQ1
 00176A A096 95 10 A CTSERR LDAA STATUS
 00177A A098 8A 20 A ORAA #\$20
 00178A A09A 97 10 A STAA STATUS
 00179A A09C 96 26 A CLRTXS LDAA CR2IMC
 00180A A09E 8A 40 A ORAA #\$40
 00181A A0A0 97 01 A STAA ADLCR2
 00182A A0A2 20 DA A07E BRA HIRQ3

00184 *READS STATUS REG 2 OF ADLC AND CHECKS FOR ERRORS
 00185 *OR IF RECEIVED FRAME WAS VALID.

Figure 11. Non-Priority Mode Program Listing (Sheet 5 of 12)

```

00187A A0A4 96 01 A RSTAT2 LDAA STATS2 GETS STATUS 2 FROM ADLC
00188A A0A6 97 2A A RSTA2R STAA SR2IMG
00189A A0A8 81 00 A CMPA #000
00190A A0AA 27 17 A0C3 BEQ RSTA4R
00191A A0AC 8A 10 A ORAA #010
00192A A0AE 1F 8A TAP
00193A A0B0 25 1B A0CD BCS ADDCK BRANCH IF ADDRESS PRESENT
00194A A0B2 29 2B A0DF BVS FRMVAL BRANCH IF RECV FRAME VALID
00195A A0B4 27 4E A104 RSTAT3 BEQ RIDLE BRANCH IF IDLE DETECTED
00196A A0B6 2B 58 A110 ABORT BRANCH IF ABORT DETECTED
00197A A0B8 96 2A A RSTA3R LDAA SR2IMG
00198A A0BA 49 ROLA
00199A A0BB 2B 0E A0CB BMI OVRUN RECEIVER OVERRUN ERROR
00200A A0BD 49 ROLA
00201A A0BE 2B 6E A12E BMI DCDERR DATA CARRIER LOST
00202A A0C0 49 ROLA
00203A A0C1 2B 5B A11E BMI FCSERR FRAME CHECK SEQUENCE ERROR
00204A A0C3 96 29 A RSTA4R LDAA SR1IMG
00205A A0C5 80 02 A SUBA #02
00206A A0C7 97 29 A STAA SR1IMG
00207A A0C9 20 B3 A07E BRA HIRQ3
00208A A0CB 20 72 A13F OVRUN BRA OVRN
00209A A0CD 17 0142 A212 ADDCK LBSR RDMAON TURN ON RECV DMA MODE
00210A A0D0 17 0089 A15C LBSR CKADD SEE IF THIS IS OUR ADDRESS
00211A A0D3 96 29 A LDAA SR1IMG
00212A A0D5 80 01 A SUBA #01 RDA IN SR1
00213A A0D7 97 29 A STAA SR1IMG
00214A A0D9 96 2A A LDAA SR2IMG
00215A A0DB 80 81 A SUBA #01 RDA AND ADD
00216A A0DD 20 C7 A0A6 BRA RSTA2R
00217A A0DF 17 00A0 A13D FRMVAL LBSR RDMAON TURN OFF RECV DMA MODE
00218A A0E2 D6 1E A LDAB CONTRL INC CONTROL NR COUNT
00219A A0E4 C4 E0 A ANDB #00 CLEAR IF 7 AND INC TO 1
00220A A0E6 C1 E0 A CMPB #00 IS IT 7 YET
00221A A0E8 27 14 A0FE BEQ FMVAL2 NO-BRANCH
00222A A0EA D6 1E A LDAB CONTRL YES-CLEAR NR COUNT TO ZERO
00223A A0EC C0 E0 A SUBB #00
00224A A0EE CB 20 A FMVAL1 ADDB #20 INC NR COUNT
00225A A0F0 D7 1E A STAB CONTRL
00226A A0F2 17 00BE A1B3 LBSR RXEND GO PREPARE FOR NEXT FRAME
00227A A0F5 96 2A A LDAA SR2IMG
00228A A0F7 17 0118 A212 LBSR RDMAON
00229A A0FA 80 02 A SUBA #02
00230A A0FC 20 A8 A0A6 BRA RSTA2R
00231A A0FE D6 1E A FMVAL2 LDAB CONTRL
00232A A100 C0 E0 A SUBB #00
00233A A102 20 EA A0EE BRA FMVAL1
00234A A104 17 0144 A24B RIDLE LBSR IDLE INDICATE THAT AN INACTIVE
00235 *IDLE WAS DETECTED
00236A A107 17 0192 A29C LBSR OUTFRM
00237A A10A 96 2A A LDAA SR2IMG
00238A A10C 80 04 A SUBA #04
00239A A10E 20 96 A0A6 BRA RSTA2R

00241A A110 17 015D A270 ABORT LBSR RABORT INDICATE AN ABORT WAS RECEIVED
00242A A113 17 0186 A29C LBSR OUTFRM
00243A A116 96 2A A LDAA SR2IMG

```

Figure 11. Non-Priority Mode Program Listing (Sheet 6 of 12)

00244A	A118	80	08	A	SUBA	#\$08	
00245A	A11A	97	2A	A	STAA	SR2IMG	
00246A	A11C	20	9A	A0B8	BRA	RSTA3R	
00247A	A11E	8D	6D	A18D	BSR	RDMAOF	TURN OFF DMA RECV MODE
00248A	A120	17	0158	A27B	LBSR	CRERR	INDICATE PCS ERROR OCCURED
00249A	A123	17	0176	A29C	LBSR	OUTFRM	
00250A	A126	96	2A	A	LDAA	SR2IMG	
00251A	A128	80	10	A	SUBA	#\$10	
00252A	A12A	97	2A	A	STAA	SR2IMG	
00253A	A12C	20	95	A0C3	BRA	RSTA4R	
00254A	A12E	8D	5D	A18D	BSR	RDMAOF	TURN OFF DMA RECV MODE
00255A	A130	17	0153	A286	LBSR	DCDLST	DATA CARRIER DETECT FROM
00256					*MODEM	WAS LOST	
00257A	A133	17	0166	A29C	LBSR	OUTFRM	
00258A	A136	96	2A	A	LDAA	SR2IMG	
00259A	A138	80	20	A	SUBA	#\$20	
00260A	A13A	97	2A	A	STAA	SR2IMG	
00261A	A13C	16	FF79	A0B8	LBRA	RSTA3R	
00262A	A13F	17	014F	A291	LBSR	OVRUN1	SET BIT TO INDICATE OVERRUN
00263A	A142	17	0157	A29C	LBSR	OUTFRM	
00264A	A145	96	2A	A	LDAA	SR2IMG	
00265A	A147	80	40	A	SUBA	#\$40	
00266A	A149	97	2A	A	STAA	SR2IMG	
00267A	A14B	15	FF6A	A0B8	LBRA	RSTA3R	

Figure 11. Non-Priority Mode Program Listing (Sheet 7 of 12)

00269 *DMA SERVICE INTERRUPT

```

00271A A14E 96 26 A HIRQ02 LDAA CR2IMG SET LAST DATA BIT IN CR2
00272A A150 8A 10 A ORAA #$10 (AUTO RESET)
00273A A152 97 01 A STAA ADLCR2
00274A A154 8D 4A A1A0 BSR TDMAOF TURN OFF DMA MODE
00275A A156 17 00CC A225 LBSR TDMAON LOAD DMA ADDRESS REG AND BCR
00276A A159 8D 7E A1D9 BSR XMIT
00277A A15B 3B RTI

```

00278 *

00279 *

00280 *

00281 *

00282 *

00283 *

00284 *

00285 *

00286 *

00287 *

00288 *

00289 *

00290 *

00291 *

00292 *

*THIS ROUTINE FETCHES THE ADDRESS FROM MEM BUFF

* AND COMPARES IT THE STATION ADDRESSES.

*IF CORRECT IT SETS THE ADD. RECV. BIT IN THE

*RXFRAM. IF NOT THIS STATIONS RECV. CLR SYNC

*IS SET AND THE RECV BEGINS LOOKING FOR THE

*FLAG CONDITON AGAIN TO SYNC ON.

```

00294A A15C 34 04 CKADD PSHB
00295A A15E D6 12 A LDAB RXFRAM FIND OUT BUFFER ADDRESS
00296A A160 C5 01 A BITB #$01
00297A A162 27 04 A168 BEQ BUFCK1
00298A A164 9E 19 A LDX RXBUF2 LOAD HIGH ADD BUFF
00299A A166 20 02 A16A BRA BUFCK2
00300A A168 9E 17 A BUFCK1 LDX RXBUF1
00301A A16A E6 84 A BUFCK2 LDAB 0,X GET ADDRESS BYTE FROM BUFFER
00302A A16C D1 1B A CMPB ADRES1 COMPARE RECV DATA TO POSSIBLE
00303A A16E 27 10 A180 BEQ CKADD2 STATION ADDRESSES
00304A A170 D1 1C A CMPB ADRES2
00305A A172 27 0C A180 BEQ CKADD2 YES-BRANCH
00306A A174 D1 1D A CMPB ADRES3
00307A A176 27 08 A180 BEQ CKADD2 YES-BRANCH
00308 * NO ADDRESS MATCH THEN CLEAR RECEIVE SYNC
00309A A178 D6 25 A LDAB CR1IMG
00310A A17A CA 20 A ORAB #$20 CLEAR SYNC IN ADLC
00311A A17C D7 00 A STAB ADLCR1 DO IT
00312A A17E 20 0A A18A BRA CKADD9
00313A A180 D6 12 A CKADD2 LDAB RXFRAM
00314A A182 CA 02 A ORAB #$02
00315A A184 D7 12 A STAB RXFRAM SET ADD BIT IN RXFRAM
00316A A186 E6 01 A LDAB 1,X GET THE CONTROL BYTE
00317A A188 D7 2B A STAB INCRCTL SAVE THE CONTROL BYTE
00318A A18A 35 04 CKADD9 PULB
00319A A18C 39 RTS

```

00321 *THIS SUBROUTINE TURNS OFF DMA CHAN 1 ENABLE AND

00322 *ADLC RECEIVE MODE OF OPERATION.

00323A A18D 34 02 RDMAOF PSHA

Figure 11. Non-Priority Mode Program Listing (Sheet 8 of 12)

PAGE 009 NOPRIORT.SA:1 NOPRI

```

00324A A18F 96 25 A LDAA CRIIMG GET IMAGE OF CRI
00325A A191 80 08 A SUBA #S08 DISABLE RX DMA MODE IN ADLC
00326A A193 97 25 A STAA CRIIMG
00327A A195 97 00 A STAA ADLCRI
00328A A197 96 54 A LDAA DMAPCR FETCH DMA PCR DATA
00329A A199 80 02 A SUBA #S02 RESET CHAN 1 ENABLE BIT
00330A A19B 97 54 A STAA DMAPCR
00331A A19D 35 02 PULA
00332A A19F 39 RTS

```

```

00334 *TURNS OFF TX DMA MODE IN ADLC AND DMA CHAN #0
00335 *IS DISABLED

```

```

00337A A1A0 34 02 TDMAOF PSHA
00338A A1A2 96 25 A LDAA CRIIMG
00339A A1A4 80 10 A SUBA #S10 RESET TXDMA BIT IN CRI
00340A A1A6 97 25 A STAA CRIIMG
00341A A1A8 97 00 A STAA ADLCRI DO IT
00342A A1AA 96 54 A LDAA DMAPCR GET PCR CONTENTS
00343A A1AC 80 01 A SUBA #S01 RESET CHAN #0 ENABLE BIT
00344A A1AE 97 54 A STAA DMAPCR DO IT
00345A A1B0 35 02 PULA
00346A A1B2 39 RTS

```

```

00348 *THIS ROUTINE LOADS THE ALTERNATE RXBUFFER ADDRESS
00349 *INTO THE DMA, CLEARS THE IN FRAME BIT, AND SETS
00350 *THE POINTER TO THE NEXT RXBUFFER AREA TO BE LOADED
00351 * INTO THE DMA

```

```

00353A A1B3 34 02 RXEND PSHA
00354A A1B5 96 12 A LDAA RXFRAM
00355A A1B7 85 80 A BITA #S80 TEST IF IN FRAME
00356A A1B9 27 1B A1D6 BEQ RXEND9 NO-BRANCH-LEAVE ROUTINE
00357A A1BB 80 86 A SUBA #S86 YES-CLEAR IN FRAME BIT & ADD & CO
00358A A1BD 85 01 A BITA #S01 TEST HI OR LO ADDRESS NEXT
00359A A1BF 27 08 A1C9 BEQ RXEND1 BRANCH TO LOAD LOW ADD
00360A A1C1 80 01 A SUBA #S01 RESET ADD START BIT IN RXFRAM
00361A A1C3 9E 17 A LDX RXBUF1 LOAD LOW ADDRESS
00362A A1C5 9F 44 A STX ADRG1H
00363A A1C7 20 06 A1CF BRA RXEND2
00364A A1C9 8B 01 A RXEND1 ADDA #S01
00365A A1CB 9E 19 A LDX RXBUF2 LOAD HIGH ADDRESS
00366A A1CD 9F 44 A STX ADRG1H
00367A A1CF 8E FFFF A RXEND2 LDX #FFFF SET UP BYTE COUNT REG TO MAX
00368A A1D2 9F 46 A STX BCRG1H
00369A A1D4 97 12 A STAA RXFRAM
00370A A1D6 35 02 RXEND9 PULA
00371A A1D8 39 RTS

```

```

00373 *SUBROUTINE TO LOAD THE TXFIFO WITH THE ADDRESS
00374 *AND CONTROL WORDS, AND TO ENABLE THE ADLC IN
00375 *THE TX DMA MODE OF OPERATION.(XMIT SECTION OF THE
00376 *ADLC IS ONCE THE INITIAL SEQUENCE HAS BEEN PERFORMED.)

```

Figure 11. Non-Priority Mode Program Listing (Sheet 9 of 12)

```

00378A A1D9 34 02          XMIT  PSHA
00379A A1DB 34 04          PSUB
00380A A1DD D6 11      A      LDAB  TXFRAM  DETERMINE WHICH TXBUF TO USE
00381A A1DF C5 01      A      BITB  #$01   IS IT #1
00382A A1E1 27 04      A1E7    BEQ    XMIT2  YES--BRANCH
00383A A1E3 9E 15      A      LDX    TXBUF2  NO---IT'S #2
00384A A1E5 20 02      A1E9    BRA    XMIT3
00385A A1E7 9E 13      A      XMIT2  LDX    TXBUF1
00386A A1E9 96 1B      A      XMIT3  LDAA   ADRES1  ADDRESS BYTE SET UP
00387A A1EB A7 84      A      STAA   0,X
00388A A1ED 30 01          INX
00389A A1EF 96 1E      A      LDAA   CONTRL  CONTROL WORD SET UP
00390A A1F1 84 0E      A      ANDA   #$0E   TEST IF 7 FRAMES SENT
00391A A1F3 81 0E      A      CMPA   #$0E
00392A A1F5 27 15      A20C    BEQ    XMTCLR  YES-BRANCH
00393A A1F7 96 1E      A      LDAA   CONTRL  NO-CONTINUE
00394A A1F9 8B 02      A      XMIT1  ADDA   #$02   INCREMENT THE NS COUNT
00395A A1FB A7 84      A      STAA   0,X    LOAD IT OUT
00396A A1FD 97 1E      A      STAA   CONTRL  SAVE THE NEW CONTROL WORD
00397A A1FF 96 25      A      LDAA   CRLIMG
00398A A201 8A 10      A      ORAA   #$10   ENABLE DMA MODE OF OPERATION
00399A A203 97 00      A      STAA   ADLCR1  DO IT
00400A A205 97 25      A      STAA   CRLIMG
00401A A207 35 04          PULB
00402A A209 35 02          PULA
00403A A20B 39          RTS
00404A A20C 96 1E      A      XMTCLR  LDAA   CONTRL
00405A A20E 80 0E      A      SUBA   #$0E   CLR NS FRAME COUNT
00406A A210 20 E7      A1F9    BRA    XMIT1

```

```

00408      *ROUTINE TO TURN RECEIVER OPERATIONS OVER TO
00409      *THE DMA CONTROLLER.

```

```

00411A A212 34 02          RDMAON  PSHA
00412A A214 96 54      A      LDAA   DMAPCR  TURN ON DMA CHAN 1 (RECV)
00413A A216 8A 02      A      ORAA   #$02
00414A A218 97 54      A      STAA   DMAPCR
00415A A21A 96 25      A      LDAA   CRLIMG  TURN ON ADLC TO DMA MODE
00416A A21C 8A 08      A      ORAA   #$08   OF OPERATION
00417A A21E 97 25      A      STAA   CRLIMG
00418A A220 97 00      A      STAA   ADLCR1  DO TI
00419A A222 35 02          PULA
00420A A224 39          RTS

```

```

00422      *SUBROUTINE TO LOAD THE ALTERNATE TX BUFFER
00423      *ADDRESS INTO THE ADDRESS REG OF THE DMA CHAN 0

```

```

00425A A225 34 02          TDMAON  PSHA
00426A A227 96 11      A      LDAA   TXFRAM  GET TX FRAME STATUS
00427A A229 85 01      A      BITA   #$01   TEST WHICH BUFFER TO USE
00428A A22B 26 08      A235    BNE    TDMON1  BRANCH NOT SET
00429A A22D 8A 01      A      ORAA   #$01
00430A A22F 9E 15      A      LDX    TXBUF2  SELECT TX BUFFER #2
00431A A231 9F 40      A      STX    ADRG0H  SET UP ADD REG IN DMA

```

Figure 11. Non-Priority Mode Program Listing (Sheet 10 of 12)

```

00432A A233 20 06 A23B BRA TDMON2
00433A A235 80 01 A TDMON1 SUBA #$01
00434A A237 9E 13 A LDX TXBUF1 SELECT TX BUFFER #1
00435A A239 9F 40 A STX ADRG0H SET UP ADD REG IN DMA
00436A A23B 97 11 A TDMON2 STAA TXFRAM
00437A A23D 8E 0400 A LDX #$0400 SET UP BCR IN DMA
00438A A240 9F 42 A STX BCRG0H DO IT
00439A A242 96 54 A LDAA DMAPCR ENABLE CHAN 0 IN DMA
00440A A244 8A 01 A ORAA #$01
00441A A246 97 54 A STAA DMAPCR
00442A A248 35 02 PULA
00443A A24A 39 RTS

```

```

00445 *SUBROUTINE TO SET THE INACTIVE IDLE BIT IN THE
00446 *STATUS SOFTWARE REGISTER

```

```

00448A A24B 34 02 IDLE PSHA
00449A A24D 96 10 A LDAA STATUS
00450A A24F 8A 01 A ORAA #$01 SET INACTIVE IDLE BIT
00451A A251 97 10 A STAA STATUS
00452A A253 35 02 PULA
00453A A255 39 RTS

```

```

00455 *SUBROUTINE TO CLEAR RX STATUS

```

```

00457A A256 96 00 A CLEAR LDAA STAT1 SAFETY CHECK OF STATUS1
00458A A258 91 2C A CMPA SRLIM2 TO MAKE SURE OF NO NEW STATUS
00459A A25A 26 07 A263 BNE NOTCLR NEW STATUS?--BRANCH
00460A A25C 96 26 A LDAA CR2IMG CLEAR RECEIVER STATUS
00461A A25E 8A 20 A ORAA #$20
00462A A260 97 01 A STAA ADLCR2
00463A A262 3B RTI
00464A A263 D6 2C A NOTCLR LDAB SRLIM2
00465A A265 53 COMB GET THE OLD STATUS
00466A A266 D7 2D A STAB SCRTCH AND COMPARE IT TO THE NEW
00467A A268 94 2D A ANDA SCRTCH GET RID OF OLD STATUS
00468A A26A 16 FE06 A073 LBRA HIRQ1 GO BACK AND SERVICE NEW STATUS
00469A A26D 16 FE03 A073 LBRA HIRQ1

```

```

00471 *SUBROUTINE TO SET THE ABORT BIT IN THE STATUS
00472 *SOFTWARE REGISTER

```

```

00474A A270 34 02 RABORT PSHA
00475A A272 96 10 A LDAA STATUS SET ABORT BIT
00476A A274 8A 02 A ORAA #$02
00477A A276 97 10 A STAA STATUS
00478A A278 35 02 PULA

```

```

00480A A27A 39 RTS
00481 *SUBROUTINE TO SET THE FCS ERROR BIT IN THE
00482 *STATUS SOFTWARE REGISTER

```

Figure 11. Non-Priority Mode Program Listing (Sheet 11 of 12)

PAGE 012 NOPRIORT.SA:1 NOPRI

```
00484A A27B 34 02      CKCERR PSHA
00485A A27D 96 10      A      LDAA      STATUS      SET FCS ERROR BIT
00486A A27F 8A 04      A      ORAA      #$04
00487A A281 97 10      A      STAA      STATUS
00488A A283 35 02      PULA
00489A A285 39      RTS
```

```
00491      *SUBROUTINE TO SET THE DCD ERROR BIT IN THE
00492      *STATUS SOFTWARE REGISTER
```

```
00494A A286 34 02      DCDLST PSHA
00495A A288 96 10      A      LDAA      STATUS
00496A A28A 8A 08      A      ORAA      #$08
00497A A28C 97 10      A      STAA      STATUS
00498A A28E 35 02      PULA
00499A A290 39      RTS
```

```
00501      *SUBROUTINE TO SET RECEIVE OVERRUN BIT IN STATUS
00502      *SOFTWARE REGISTER AND CLEAR THE RECEIVER STATUS
```

```
00504A A291 34 02      OVRUN1 PSHA
00505A A293 96 10      A      LDAA      STATUS      SET RX OVERRUN BIT IN STATUS
00506A A295 8A 10      A      ORAA      #$10
00507A A297 97 10      A      STAA      STATUS
00508A A299 35 02      PULA
00509A A29B 39      RTS
```

```
00511      * SUBROUTINE TO REMOVE THE ADDRESS,
00512      * CONTROL, AND IN FRAME BITS FROM THE
00513      * SOFTWARE REGISTER RXFRAM.
```

```
00515A A29C 34 02      OUTFRM PSHA
00516A A29E 96 12      A      LDAA      RXFRAM
00517A A2A0 85 80      A      BITA      #$80      CK IF IN FRAME
00518A A2A2 27 04      A2A8    BEQ      OUTFM9    NO-BRANCH
00519A A2A4 80 86      A      SUBA      #$86      YES-DECLARE END OF FRAME
00520A A2A6 97 12      A      STAA      RXFRAM    SAVE IT
00521A A2A8 35 02      OUTFM9 PULA
00522A A2AA 39      RTS
00523      END
```

```
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000
```

Figure 11. Non-Priority Mode Program Listing (Sheet 12 of 12)

Figure 11. Non-Exhaustive Mode Program Listing (Sheet 12 of 12)

PAGE 012 NOPRIORT.RA:1 NOPRI											
00484A	A278	34	02	00484A	A278	34	02	00484A	A278	34	02
00485A	A27D	9C	18	00485A	A27D	9C	18	00485A	A27D	9C	18
00486A	A27E	8A	04	00486A	A27E	8A	04	00486A	A27E	8A	04
00487A	A281	97	18	00487A	A281	97	18	00487A	A281	97	18
00488A	A283	35	02	00488A	A283	35	02	00488A	A283	35	02
00489A	A285	38		00489A	A285	38		00489A	A285	38	
*SUBROUTINE TO SET THE DCD ERROR BIT IN THE											
*STATUS SOFTWARE REGISTER											
00491				00491				00491			
00492				00492				00492			
00493A	A288	34	02	00493A	A288	34	02	00493A	A288	34	02
00494A	A288	9C	18	00494A	A288	9C	18	00494A	A288	9C	18
00495A	A28A	8A	08	00495A	A28A	8A	08	00495A	A28A	8A	08
00497A	A28C	97	18	00497A	A28C	97	18	00497A	A28C	97	18
00498A	A28E	35	02	00498A	A28E	35	02	00498A	A28E	35	02
00499A	A290	38		00499A	A290	38		00499A	A290	38	
*SUBROUTINE TO SET RECEIVE OVERRUN BIT IN STATUS											
*SOFTWARE REGISTER AND CLEAR THE RECEIVER STATUS											
00501				00501				00501			
00502				00502				00502			
00503A	A291	34	02	00503A	A291	34	02	00503A	A291	34	02
00505A	A293	9C	18	00505A	A293	9C	18	00505A	A293	9C	18
00506A	A295	8A	08	00506A	A295	8A	08	00506A	A295	8A	08
00507A	A297	97	18	00507A	A297	97	18	00507A	A297	97	18
00508A	A299	35	02	00508A	A299	35	02	00508A	A299	35	02
00509A	A29B	38		00509A	A29B	38		00509A	A29B	38	
*SUBROUTINE TO REMOVE THE ADDRESS,											
*CONTROL, AND IN FRAME BITS FROM THE											
*SOFTWARE REGISTER RXPTRM.											
00511				00511				00511			
00512				00512				00512			
00513				00513				00513			
00514A	A29C	34	02	00514A	A29C	34	02	00514A	A29C	34	02
00515A	A29E	9C	18	00515A	A29E	9C	18	00515A	A29E	9C	18
00517A	A2A6	82	08	00517A	A2A6	82	08	00517A	A2A6	82	08
00518A	A2A7	37	04	00518A	A2A7	37	04	00518A	A2A7	37	04
00519A	A2A4	8A	08	00519A	A2A4	8A	08	00519A	A2A4	8A	08
00520A	A2A6	97	18	00520A	A2A6	97	18	00520A	A2A6	97	18
00521A	A2A8	35	02	00521A	A2A8	35	02	00521A	A2A8	35	02
00522A	A2AA	38		00522A	A2AA	38		00522A	A2AA	38	
00523				00523				00523			
TOTAL WARNINGS 00000--00000											
TOTAL ERRORS 00000--00000											
00524A	A2A9	34	02	00524A	A2A9	34	02	00524A	A2A9	34	02
00525A	A2AB	9C	18	00525A	A2AB	9C	18	00525A	A2AB	9C	18
00527A	A2AB	82	08	00527A	A2AB	82	08	00527A	A2AB	82	08
00528A	A2A7	37	04	00528A	A2A7	37	04	00528A	A2A7	37	04
00529A	A2A4	8A	08	00529A	A2A4	8A	08	00529A	A2A4	8A	08
0052AA	A2A6	97	18	0052AA	A2A6	97	18	0052AA	A2A6	97	18
0052BA	A2A8	35	02	0052BA	A2A8	35	02	0052BA	A2A8	35	02
0052CA	A2AA	38		0052CA	A2AA	38		0052CA	A2AA	38	
0052D				0052D				0052D			
0052EA	A2AB	34	02	0052EA	A2AB	34	02	0052EA	A2AB	34	02
0052FA	A2AD	9C	18	0052FA	A2AD	9C	18	0052FA	A2AD	9C	18
00530A	A2AD	82	08	00530A	A2AD	82	08	00530A	A2AD	82	08
00531A	A2A7	37	04	00531A	A2A7	37	04	00531A	A2A7	37	04
00532A	A2A4	8A	08	00532A	A2A4	8A	08	00532A	A2A4	8A	08
00533A	A2A6	97	18	00533A	A2A6	97	18	00533A	A2A6	97	18
00534A	A2A8	35	02	00534A	A2A8	35	02	00534A	A2A8	35	02
00535A	A2AA	38		00535A	A2AA	38		00535A	A2AA	38	
00536				00536				00536			
00537A	A2AB	34	02	00537A	A2AB	34	02	00537A	A2AB	34	02
00538A	A2AD	9C	18	00538A	A2AD	9C	18	00538A	A2AD	9C	18
00539A	A2AD	82	08	00539A	A2AD	82	08	00539A	A2AD	82	08
0053AA	A2A7	37	04	0053AA	A2A7	37	04	0053AA	A2A7	37	04
0053BA	A2A4	8A	08	0053BA	A2A4	8A	08	0053BA	A2A4	8A	08
0053CA	A2A6	97	18	0053CA	A2A6	97	18	0053CA	A2A6	97	18
0053DA	A2A8	35	02	0053DA	A2A8	35	02	0053DA	A2A8	35	02
0053EA	A2AA	38		0053EA	A2AA	38		0053EA	A2AA	38	
0053F				0053F				0053F			
00540A	A2AB	34	02	00540A	A2AB	34	02	00540A	A2AB	34	02
00541A	A2AD	9C	18	00541A	A2AD	9C	18	00541A	A2AD	9C	18
00542A	A2AD	82	08	00542A	A2AD	82	08	00542A	A2AD	82	08
00543A	A2A7	37	04	00543A	A2A7	37	04	00543A	A2A7	37	04
00544A	A2A4	8A	08	00544A	A2A4	8A	08	00544A	A2A4	8A	08
00545A	A2A6	97	18	00545A	A2A6	97	18	00545A	A2A6	97	18
00546A	A2A8	35	02	00546A	A2A8	35	02	00546A	A2A8	35	02
00547A	A2AA	38		00547A	A2AA	38		00547A	A2AA	38	
00548				00548				00548			
00549A	A2AB	34	02	00549A	A2AB	34	02	00549A	A2AB	34	02
0054AA	A2AD	9C	18	0054AA	A2AD	9C	18	0054AA	A2AD	9C	18
0054BA	A2AD	82	08	0054BA	A2AD	82	08	0054BA	A2AD	82	08
0054CA	A2A7	37	04	0054CA	A2A7	37	04	0054CA	A2A7	37	04
0054DA	A2A4	8A	08	0054DA	A2A4	8A	08	0054DA	A2A4	8A	08
0054EA	A2A6	97	18	0054EA	A2A6	97	18	0054EA	A2A6	97	18
0054FA	A2A8	35	02	0054FA	A2A8	35	02	0054FA	A2A8	35	02
00550A	A2AA	38		00550A	A2AA	38		00550A	A2AA	38	
00551				00551				00551			
00552A	A2AB	34	02	00552A	A2AB	34	02	00552A	A2AB	34	02
00553A	A2AD	9C	18	00553A	A2AD	9C	18	00553A	A2AD	9C	18
00554A	A2AD	82	08	00554A	A2AD	82	08	00554A	A2AD	82	08
00555A	A2A7	37	04	00555A	A2A7	37	04	00555A	A2A7	37	04
00556A	A2A4	8A	08	00556A	A2A4	8A	08	00556A	A2A4	8A	08
00557A	A2A6	97	18	00557A	A2A6	97	18	00557A	A2A6	97	18
00558A	A2A8	35	02	00558A	A2A8	35	02	00558A	A2A8	35	02
00559A	A2AA	38		00559A	A2AA	38		00559A	A2AA	38	
0055A				0055A				0055A			
0055BA	A2AB	34	02	0055BA	A2AB	34	02	0055BA	A2AB	34	02
0055CA	A2AD	9C	18	0055CA	A2AD	9C	18	0055CA	A2AD	9C	18
0055DA	A2AD	82	08	0055DA	A2AD	82	08	0055DA	A2AD	82	08
0055EA	A2A7	37	04	0055EA	A2A7	37	04	0055EA	A2A7	37	04
0055FA	A2A4	8A	08	0055FA	A2A4	8A	08	0055FA	A2A4	8A	08
00560A	A2A6	97	18	00560A	A2A6	97	18	00560A	A2A6	97	18
00561A	A2A8	35	02	00561A	A2A8	35	02	00561A	A2A8	35	02
00562A	A2AA	38		00562A	A2AA	38		00562A	A2AA	38	
00563				00563				00563			
00564A	A2AB	34	02	00564A	A2AB	34	02	00564A	A2AB	34	02
00565A	A2AD	9C	18	00565A	A2AD	9C	18	00565A	A2AD	9C	18
00566A	A2AD	82	08	00566A	A2AD	82	08	00566A	A2AD	82	08
00567A	A2A7	37	04	00567A	A2A7	37	04	00567A	A2A7	37	04
00568A	A2A4	8A	08	00568A	A2A4	8A	08	00568A	A2A4	8A	08
00569A	A2A6	97	18	00569A	A2A6	97	18	00569A	A2A6	97	18
0056AA	A2A8	35	02	0056AA	A2A8	35	02	0056AA	A2A8	35	02
0056BA	A2AA	38		0056BA	A2AA	38		0056BA	A2AA	38	
0056C				0056C				0056C			
0056DA	A2AB	34	02	0056DA	A2AB	34	02	0056DA	A2AB	34	02
0056EA	A2AD	9C	18	0056EA	A2AD	9C	18	0056EA	A2AD	9C	18
0056FA	A2AD	82	08	0056FA	A2AD	82	08	0056FA	A2AD	82	08
00570A	A2A7	37	04	00570A	A2A7	37	04	00570A	A2A7	37	04
00571A	A2A4	8A	08	00571A	A2A4	8A	08	00571A	A2A4	8A	08
00572A	A2A6	97	18	00572A	A2A6	97	18	00572A	A2A6	97	18
00573A	A2A8	35	02	00573A	A2A8	35	02	00573A	A2A8	35	02
00574											

